

Estimation of Relative Position between Real Objects using Sensor Nodes

Yutaka Yanagisawa
NTT Communication Science
Laboratories
yutaka@cslab.kecl.ntt.co.jp

Takuya Maekawa
NTT Communication Science
Laboratories
maekawa@cslab.kecl.ntt.co.jp

Takeshi Okadome
NTT Communication Science
Laboratories
houmi@idea.brl.ntt.co.jp

ABSTRACT

In this paper, we describe the challenge of estimating the *relative position* between objects using a low-cost positioning device with a simple physical restriction. Also, we show the result of our experiment to examine the accuracy in estimation of the relative position.

Categories and Subject Descriptors

J.m [Computer Applications]: Miscellaneous

General Terms

Algorithms

Keywords

Relative Positioning, Ubiquitous Computing

1. INTRODUCTION

In recent years, many ubiquitous application systems have provided services based on various types of sensor data from small sensor nodes in the real world. The current position of an object is one of the most significant bits of data used in such application systems [1], including those providing context-aware services and location-aware services [4] [2] [5] [7]. A system for obtaining an object's position is generally called a positioning system. Most existing positioning systems obtain an object's position data as an *absolute* position $\langle x, y, z \rangle$ on a global coordinate system. Obviously, absolute positions can be applied to any service using position data; however, a *relative* position expression is more familiar to users for understanding object location. For example, in an object search service, the query "where is my book?" requires the book's position. We usually prefer a *relative* position answer, such as "the book is on the desk in this room," to an *absolute* position answer like "the book is at $\langle x, y, z \rangle$." In ubiquitous application systems that deal with the position information of objects, users want to represent the condition of positions as *relative* information among several objects. We call this relative information the *relative* position between objects.

Application systems can determine relative position using the accurate absolute position obtained by existing positioning devices.

In general, we must use high cost sensors embedded in an environment to obtain accurate absolute position information. For example, GPS receivers can accurately track an object's position, but a highly accurate receiver is much more expensive than less accurate receivers. An image-based tracking system must use many cameras to capture accurate images of each object to avoid occlusion.

For handling only accurate *relative* position between objects, however, relative position can be estimated by physical conditions, sensor data, and inaccurate absolute positions obtained by low-cost devices. Thus, in this paper we propose a method to estimate accurate relative positions among objects using only low-cost devices.

Before explaining the estimation method, we define relative position. As mentioned above, many context-aware applications and activity recognition systems have adopted relative position among objects, for example, "the book is *on* your desk" and "a person moved *into* the office," but few researchers have formally defined relative position. We found a hint to defining relative position in cognitive linguistics research. Lakoff and Johnson mentioned [3] that humans tend to use comparable spatial and physical relations to represent objects and abstract concepts, for instance, big-small, in-out, far-near, and up-down. They argued that spatial relations (up-down, left-right, and front-back) are often used to represent relations among these issues. Based on their research, in this paper we focused on estimating relative spatial relations. A detailed discussion of relative position is given in Section 3.

After the discussion, we describe our challenge for estimating relative positions among objects using both sensor data and inaccurate absolute position in Section 4. Section 5 describes the results of the relative position estimation of five real objects put in an experiment field. In this experiment, we used low-cost Zigbee wireless communication devices to specify the absolute positions of each object. Each object has our developed small computer device, which includes an acceleration sensor, a direction sensor, and a Zigbee device to translate sensor data to the host computer. We also calculate relative position to compare the performance of each method using IS-600mk2, a highly accurate ultrasonic positioning system. Finally, we show that our estimation method has better accuracy than the highly accurate system in several cases.

2. RELATED WORK

Position information processing is a significant technique in various services provided by a ubiquitous computing environment. For instance, Schiele and Antifakos [6] proposed a proximity detection system that detects dangerous situations involving chemicals. The collection of object positions obtained by positioning devices can determine whether dangerous chemicals are too close together. Lampe et al. [4] proposed a ubiquitous service for aircraft maintenance, where each tool has a sensor node that can obtain the lo-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICPS'08, July 6-10, 2008, Sorrento, Italy.

Copyright 2008 ACM 978-1-60558-135-4/08/07 ...\$5.00.

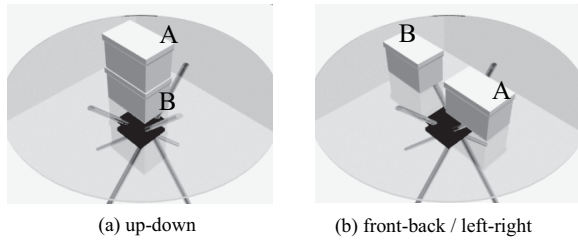


Figure 1: Examples of relative positions

cation of other nodes. Their service system can detect necessary tools and can determine whether the tools for a maintenance process are assembled with position data obtained from sensor nodes. Hightower et al. [2] proposed a personal mobile device that automatically learns where users go and detects when they return. In a health-care service presented by Lester et al. [5], the service system recognizes each person’s activity using position data. Each of these systems uses the location information about objects and users to provide services in ubiquitous and mobile computing environments. Event description with conditions corresponding to positions is a significant technique in such application systems.

Basically, their system focused on providing rule-based ubiquitous services rather than the management of consistency among described rules. Users can freely describe an event as a rule without strong restrictions. As mentioned in Section 1, these existing approaches provide simple services well with a few rule descriptions. To provide large-scale services using many complicated rules, we must manage consistency because the conflicts in ambiguous rules confuse users in the description of events that provide services. Through the history of research in programming languages, many researchers and programmers have recognized that such inconsistency in rule description causes serious problems when providing stable services. To further develop rule-based ubiquitous services, we now discuss and build a consistent model that can describe events using position data, even if we have to abandon convenience.

3. RELATIVE POSITION

Cognitive linguistics research argues that humans grasp an abstract concept as a *metaphor* that is represented as a physical object. In [3], Lakoff and Johnson state that an ‘orientational metaphor’ is used for handling a relationship between abstract concepts, or a situation of a concept. An orientational metaphor is based on the spatial directions of an object in a space. They also cite seven concrete spatial relations: up-down, in-out, on-off, front-back (left-right), far-near, deep-shallow (high-low), and central-peripheral. Metaphors including either up-down or in-out relations are especially frequent in daily conversations. Their spatial relations argument can be applied to define relative positions, which application systems use as position information.

The four relations up-down, in-out, on-off, and front-back (left-right) share the following feature: a relation between two objects can only be determined as one situation with each object’s relative position. For example, when looking at Figure 1(a), we can easily determine the up-down relation between *A* and *B* with each object’s relative position, such that “*A* is higher than *B*.” Similar to the up-down relation, we can give a unique representation about the other relation. We call these spatial relations “directional relations.” Note that both right-left and front-back relations seem ambiguous, which probably reflects the different viewpoints of each user. When user *X* stands in front of box *A* in Figure 1(b) and another user *Y*

Table 1: Relative position functions

Relation	x_1	x_2	x_3	x_4
R_{up}	up	down	-	-
R_{dir}	front	back	left	right

looks at the area from the left side, user *X* says box *A* is in front of box *B*; however, user *Y* says box *A* is on the left side of box *B*. We can avoid such ambiguity with a fixed viewpoint.

On the other hand, for relations in the latter groups, the “distance-based relations,” we do not always give a unique relation between two objects without ambiguity. For instance, we do not determine the far-near relation between *A* and *B* because each human looking at this situation has a different subjective perspective about the distance. Even if a person says “*A* is close to *B*,” another person may say, “I think *A* is a bit far from *B*.” In other words, deciding the distance-based relation between two objects is more complicated than deciding the directional relation.

As a first step to estimating relative positions, this paper describes a method to estimate four directional relations: in-out, on-off, up-down, and left-right (front-back).

Here, we formally define each relative position as function R . We denote the solid object as o , and the set of objects is denoted as $O = o_1, o_2, \dots, o_n$. The shape of the solid object never changes, and the space occupied by a solid object shares no space with any other solid objects. Relative position function R can be defined: given two solid objects $o_1, o_2 \in O$ and $X = \{x_1, x_2, \dots, x_n, \phi\}$, relation function $R : O \times O \rightarrow X$ is $R(o_1, o_2) = \{x_1, x_2, \dots, \}$. Value set X is given for each relation R , shown in Table 1.

4. ESTIMATION

In this section, we describe our approach to estimating relative positions between two objects using low-cost positioning devices.

In our experiments, we use a positioning device whose error is less than 50 cm based on a Zigbee wireless device for estimation of relative positions among the objects whose size is almost equal to 50cm. Because it is difficult to estimate relative position between objects whose size is much less than the error of the positioning devices, we try to estimate the relative positions using physical restrictions in the case where the size of an object is less than the error as our first challenge. We adopt two physical restrictions such that “An object never exists in the air” and “An object never overlaps with any other object.” We aim to examine the accuracy of our method to estimate relative positions using both the physical restrictions and low-cost Zigbee devices.

In order to satisfy the second restriction strictly, we must introduce complicated procedures into the estimation method. Here, our method adopts simple estimation procedures that solve the conflicts only on the horizontal axis. In the rest of this section, we show the simple estimation procedure.

Before the explanations of the procedure, we mention the approximated sphere. To simplify the problem to estimate relative position, we introduce an approximated sphere s into each object o_1, \dots, o_n . First, we define the radius of the inscribed sphere of an object o as $r_{in}(o)$. Also we denote the circumscribed sphere of o as $r_{circum}(o)$. Then a radius r_s of approximated sphere s is given as the equation: $r_s = (r_{in}(o) + r_{circum}(o))/2$. In this estimation procedure, we replace real objects with these approximated spheres and the estimation system tries to solve the conflicts between the spheres.

Our preliminary preparation of the estimation shows good accu-

racy even if the radius r_s is the same length as the inscribed sphere for o in the case that the shape of the object is similar to a sphere. On the other hand, we find errors over our tolerance in several objects whose shape is not similar to a sphere, for example, a vase, a door, a display, and so on. This is why we introduce the previously defined approximated sphere in our experiments. Note that the center point of s is denoted as $p(s) = (x, y, z)$. The positioning system can calculate the position $p(s_i)$ for o_i using the obtained position of o_i and the shape information of o_i .

In the rest of this section, we show the simple estimation procedure.

1. Preparation. For all objects $o_1, \dots, o_n \in O$ existing in an observed field, the system determined the positions of each approximated sphere s_1, \dots, s_n . When an object o is added in the field, the system calculates the distance $d(s, s_i)$ between the approximated sphere s for o and s_i for $\forall o_i \in O$. For 'ground' object, we must consider a distance $d(s, G)$ between s and the ground G , which has the following three features: the ground is an unmovable object, the shape of the ground is a horizontal plane, and the ground has no overlaps with any other objects. We define $d(s, G) = r_s$ based on these features.
2. Resolution of 'flying' objects. If the system does not find any conflicts between s and s_i for $i \in O$, such that $d(s, s_i) > r_s + r_{s_i}$, the system determines the position of o at the position where the object does not 'fly' without moving the vertical position. In other words, the system moves the object down by the horizontal axis to solve the problem that the object does not satisfy the first restriction.
3. Resolution of conflicts between objects. When the system finds overlaps between s and another sphere s_i , the system moves the sphere s up by the horizontal axis at the point where these objects have no overlaps, i.e., $d(s, s_i) = r_s + r_{s_i}$. After this process, if the system has another conflict, the system repeats this process until solving all conflicts.

After these processes determine the position $p(s)$, the system estimates the relative position between o and a given object $o_j \in O$.

- R_{up} : When the height of o is higher than the height of objects o_j , the system estimates $R_{up}(o, o_j) = \text{up}$ and $R_{up}(o_j, o) = \text{down}$. If the height of o is equal to the height of o_j , we define $R_{up}(o, o_j) = R_{up}(o_j, o) = \text{up}$.
- R_{dir} : For estimating $R_{dir}(o, o_j)$, the system must fix the user's viewpoint at $vp = (x_v, y_v, z_v)$, where no spheres include the point vp . Given a fixed viewpoint vp , the system can determine R_{dir} by comparing horizontal positions of both objects.

As mentioned, because our procedure does not solve the horizontal overlaps among spheres and also our method approximates the shape of an object as a sphere, the complexity of calculation is much less than the algorithm to satisfy the restriction strictly. The complexity of the procedure to solve vertical overlaps is $O(n)$ for objects o_1, \dots, o_n and the complexity of estimation is $O(1)$. This simplicity has much advantage over other positioning devices for implementation on small sensor nodes.

5. EXPERIMENT

In this section, we describe three experiments to estimate relative position with an inaccurate positioning system.

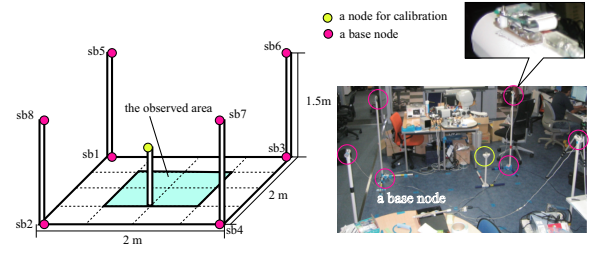


Figure 2: Experimental field

Table 2: Accuracy of estimation and positioning

patterns	Found relations			Zigbee		IS600mk2		occ.
	up	front	left	R_{up}	R_{dir}	R_{up}	R_{dir}	
I	1	0	0	1	-	1	-	0
II	2	0	1	1	1	1	1	0
III	3	2	1	1	.33	1	1	0
IV	4	4	2	1	.83	1	1	0
V	3	0	0	1	-	.33	-	1
VI	6	0	0	1	-	.20	-	2
VII	10	0	0	.80	-	.16	-	1
VIII	4	0	2	.75	1	.50	1	2
IX	6	0	4	.83	.75	.33	1	3

1. We have an experiment to estimate relative positions (up-down, front-back, and left-right) with our proposed method. In this experiment, we adopt a Zigbee device to obtain positions. This positioning system has within-50cm error.
2. To compare our method with a highly accurate positioning system, we use ultrasonic positioning system IS-600mk2 to obtain positions. This system has error of less than 2cm.
3. As an additional experiment, we calculate the position $\langle x, y, z \rangle$ using the results of estimation in experiment 1.

5.1 Experimental Settings

The experiment field is shown in Figure 2. This field has 8 base nodes sb_1, \dots, sb_8 , and each has a Zigbee device. We have four objects to settle in the field, and each object has a small computer board, which has both a 10 bit 3-way acceleration sensor to detect the gravity and 2-way geomagnetic sensor to detect north. A small computer can obtain sensor data per 1/30 msec, moreover, the computer can send data to a host computer through the Zigbee device. The estimation system can know the different angle between the local coordinate system of an object and the global coordinate system. The acceleration sensor has error within 2 degrees, and the geomagnetic sensor has error within 1 degree.

Figure 3 shows 9 arrangement patterns of objects used in the experiments. The first object is put onto the floor as pattern I, and the next object is added as pattern II or V. When an object has been added and stopped, we try to estimate relative positions.

5.2 Positioning

Here, we explain our positioning method using Zigbee devices. The field has eight base nodes. The position of each node is given as $(1, 1, 0)$, $(1, -1, 0)$, $(-1, -1, 0)$, $(-1, 1, 0)$, $(1, 1, 1.5)$, $(1, -1, 1.5)$, $(-1, -1, 1.5)$, and $(-1, 1, 1.5)$. A Zigbee device can know the power level of the radio frequency from the other Zigbee device, but each node has a different power level, so that we must check this. For the calibration, we obtain the power level on each point on the matrix divided by 50cm in the field $(-1, -1, 0) - (1, 1, 1.5)$.

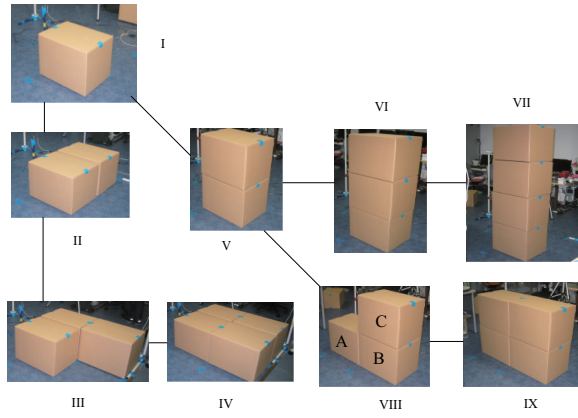


Figure 3: Arrangement patterns

Given constraints a_1, \dots, a_8 , power levels r_1, \dots, r_8 and position vector $\vec{v}_1 = (-1, -1, 0), \dots, \vec{v}_8 = (1, 1, 1.5)$ of each base node, the position \vec{p} can be calculated with the following equation:

$$\vec{p} = (x, y, z) = \sum_{i=1}^8 a_i r_i \vec{v}_i.$$

To obtain a list of a_i , we make a list of power level r_i in the experiment where each object is put at each point on the matrix, which is divided into $50\text{cm} \times 50\text{cm}$ squares, as shown in Figure 2. Because each \vec{v}_i and p are given in this experiment, we can obtain a list of a_i using the previous equation.

Based on this method, the range of the positioning error is between 30cm and 1m. In our estimation experiment, we put objects near more accurate base nodes, which are sb_1, sb_2, sb_5 , and sb_6 in Figure 2. Within the area surrounded by bold lines in the figure, the maximum error is less than 60cm. There are many other positioning methods using a Zigbee device, and these methods can also be applied to our proposed method. Many of them, however, need more complex settings or calibration. We adopt a simple positioning method in our experiment because our goal is estimating relative position.

Additionally, we explain the ultrasonic positioning device INTERSENSE IS600mk2. We set the antenna of this system on the ceiling of the experimental field and we put beacons on four objects. This system can obtain the position of the beacon using the antenna with only 2cm positioning error as long as the antenna can see each beacon, in other words, the field has no occlusion problem.

5.3 Results

Table 2 shows the accuracy of the estimation process. For each setting pattern, we compare the result of the estimation with the real relation between two objects. The columns in the found relation show a number of relations included in each pattern. For example, pattern II has three objects (A, B and ground) so that the estimation system must find 2 up relations $R_{up}(A, \text{ground}) = \text{up}$ and $R_{up}(B, \text{ground}) = \text{up}$. When the accuracy of this experiment is 0.67 for R_{up} , the system finds two correct results about R_{up} and one incorrect estimation. For instance, the value at the cell in the column of IS600mk2's R_{up} and the row of pattern IX means that the system finds only a correct relation $R_{up}(A, \text{ground}) = R_{up}(C, \text{ground}) = \text{up}$ but the system fails to estimate the relation $R_{up}(C, A) = R_{up}(C, B) = R_{up}(B, \text{ground}) = R_{up}(A, B) = \text{up}$. When the antenna cannot detect a beacon, we consider the estimation failed in the case of the pattern V–X. For this consideration,

the table shows a number of objects for which the antenna cannot detect the beacon. Obviously, the estimation error has a relation with a number of objects that cannot be detected by the device.

As an overview, the accuracy in estimation of the up-down relation is much better than the ultrasonic system in several cases where the system has occlusions.

6. CONCLUSIONS

In this paper, we described the challenge of estimating relative position using a low-cost positioning device and physical restrictions. The results of our proposed estimation method have almost the same accuracy as the highly accurate system in several cases. We do not consider our proposed system can replace existing positioning systems, but our system can enhance the existing system for detecting relative position between objects. In the future, we will improve our method to obtain more accurate relative position than the current method.

7. REFERENCES

- [1] J. Hightower and G. Borriella. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, 2001.
- [2] J. Hightower, S. Consolvo, A. LaMarca, I. E. Smith, and J. Hughes. Learning and recognizing the places we go. In *International Conference on Ubiquitous Computing (Ubicomp)*, pages 159–176, 2005.
- [3] G. Lakoff and M. Johnson. *Metaphors We Live By*. University Of Chicago Press, 1980.
- [4] M. Lampe, M. Strassner, and E. Fleisch. A ubiquitous computing environment for aircraft maintenance. In *ACM symposium on Applied computing (SAC '04)*, pages 1586–1592, New York, NY, USA, 2004. ACM Press.
- [5] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. In *International Conference on Pervasive Computing*, pages 1–16, 2006.
- [6] B. Schiele and S. Antifakos. Beyond position awareness, 2001.
- [7] T. Terada, M. Tsukamoto, K. Hayakawa, T. Yoshihisa, Y. Kishino, A. Kashitani, and S. Nishio. Ubiquitous chip: A rule-based i/o control device for ubiquitous computing. In *International Conference on Pervasive Computing*, pages 238–253, 2004.