The Lighthouse Location System for Smart Dust[1]

Kay Römer

Department of Computer Science
ETH Zurich, Switzerland
`roemer@inf.ethz.ch`

Abstract

Smart Dust sensor networks – consisting of cubic millimeter scale sensor nodes capable of limited computation, sensing, and passive optical communication with a base station – are envisioned to fulfill complex large scale monitoring tasks in a wide variety of application areas. In many potential Smart Dust applications such as object detection and tracking, fine-grained node localization plays a key role. However, due to the unique characteristics of Smart Dust, traditional localization systems cannot be used. In this paper we present and analyse the Lighthouse location systems, a novel laser-based location system for Smart Dust, which allows tiny dust nodes to autonomously estimate their location with high accuracy without additional infrastructure components besides a modified base station device. Using an early 2D prototype of the system, node locations could be estimated with an average accuracy of about 2% and an average standard deviation of about 0.7% of the node's distance to the base station.

1 Introduction

Wireless sensor networks (WSN) [1] are currently an active field of research. A WSN consists of large numbers of co-operating small-scale nodes capable of limited computation, wireless communication, and sensing. In a wide variety of application areas including geophysical monitoring, precision agriculture, habitat monitoring, transportation, military systems and business processes, WSNs are envisioned to fulfill complex monitoring tasks.

In many typical sensor network applications, fine-grained physical locations of individual sensor nodes play an important role. Examples include target detection (*where* is the target?), target tracking (*where* and *how fast* is a target moving?), and target classification (what are *size* and *shape* of the target?). Moreover, *location-dependent queries* can increase both the utility and the lifetime of a sensor network. By directing a query only to nodes in a certain geographical region or by making certain query parameters (e.g., sensor sampling rate) a function of the node location, valuable energy resources can be saved by restricting the sensor network activity to what is actually needed to answer a query.

Techniques for physical location sensing have been studied for a long time, among others, in the context of mobile computing systems [16]. More recently, some of the approaches developed there have been adopted for WSN [3, 4, 6, 10, 13, 25], mainly focusing on systems based on certain characteristics such as time-of-flight, received signal strength, signal range of ultrasound and radio waves. This adoption is often possible, because in many respects a wireless sensor node is not too much different from a mobile computing device like a PDA with WLAN access. Compared to a PDA, however, sensor nodes have rather limited resources due to their required small size and cost. Nevertheless it is often possible (both energy-wise and size-wise) to equip such sensor nodes with low power radios or small ultrasound transducers as enablers for location sensing systems.

However, research is already on the way to create the next generation of sensor nodes, for example at UC Berkeley [19, 31]. Due to their envisioned cubic-millimeter size, they are called "Smart Dust". By making nodes inexpensive and easy-to-deploy, Smart Dust opens up new applications areas. The radical size reduction mandates a revolutionary change in the used communication technology when compared to current WSN technology. Traditional radio technology presents a problem because Smart Dust nodes offer very limited space for antennas. Furthermore, radio transceivers are relatively complex circuits, making it difficult to reduce their power consumption to the level required by Smart Dust. In order to meet these requirements, [19] suggests the use of laser-based free-space optical transmission. However, due to power restrictions, near future Dust nodes will most likely make use of passive optical communication only, limiting communication to a bidirectional link between a base station device and each node.

This revolutionary new technology presents a whole new set of challenges to location sensing systems. Traditional systems based on radio waves and ultrasound are ruled out due to their power consumption and size requirements. The expected unprecedented scale of Smart Dust deployments will further challenge a location system.

In this paper, we present the Lighthouse location system for future WSN systems that are similar to the early Smart Dust prototypes developed at UC Berkeley [19]. By extending the base station, this system allows Smart Dust to autonomously estimate their physical location with respect to a base station with high precision over distances of tens of meters without node calibration. Besides a modified base station, the system does not require any additional infrastructure components. This is achieved by a new cylindrical lateration method. In contrast to traditional spherical methods, this approach does not have a wide baseline requirement (see Section 3). On the receiver side, only a simple optical receiver (amplified photo diode), moderate processing capabilities, and little memory are needed. That is, only marginal changes to the Smart Dust prototype developed at UC Berkeley are necessary.

We first describe future Smart Dust systems and compare them to more traditional commercial-off-the-shelf (COTS) sensor nodes. We will then describe the challenges of a location system for Smart Dust, before presenting the Lighthouse location system itself. The latter includes a description of the basic approach, the presentation of a prototype system, a set of initial measurements, and a first analysis of several system aspects. We conclude the paper with mentioning related work, our current work, and future research directions.

2 Smart Dust

As described in [19], Smart Dust nodes as envisioned by the Berkeley Smart Dust project will consist of a small battery, a solar cell, a power capacitor, sensors, a processing unit, an optical receiver, and a corner-cube retroreflector (CCR) within a space of about one cubic millimeter. Later versions might also contain an active transmitter based on a semiconductor laser diode. However, the high power consumption of the laser diode significantly limits the usefulness of such a component. Therefore, in the near future, communication will be possible only between sensor nodes and a so-called base station transceiver (BST).

The BST mainly consists of a steerable laser and a compact imaging receiver. For downlink communication, the BST points the modulated laser beam at the optical receiver of a node. For uplink communication, the BST points an unmodulated laser beam at the node, which modulates the laser beam and reflects it back to the BST using its CCR. Using its imaging receiver, the BST can receive and decode transmissions from dust nodes.

Obviously, this communication scheme requires an uninterrupted line-of-sight path. For many of the environmental monitoring applications envisioned for Smart Dust, however, this is not a major problem. Additionally, communication is only possible if the node's optical receiver and CCR point to the BST, so that only a fraction of deployed nodes will be able to communicate. This should not be a problem, however, if the dust node density is high enough.

Having these Smart Dust characteristics in mind, what are the differences to state-of-the-art RF-based WSN with respect to location sensing? The main differences clearly stem from the tremendous size reduction from several cubic centimeters to a few cubic millimeters. The small size also imposes tight limits on the available energy, which in turn restricts communication, memory, and processing capabilities of dust nodes. Another difference is caused by the passive optical communication scheme of dust nodes, making near future Smart Dust systems essentially single-hop networks without direct node-to-node communication. We can summarize the differences between current WSN and future Smart Dust systems as follows with respect to location systems:

- Small size: current RF antennas for radio waves and transducers for ultrasound are too large for dust nodes.

- Mobility: future Smart Dust nodes are likely to be small enough to be moved by winds or even to remain suspended in air, buoyed by air currents. This is in contrast to current sensor nodes, which are typically immobile due to their size and weight.

- Large scale: the expected small size and low cost of future Smart Dust nodes will allow very large scale deployments in terms of the number of nodes.

- Limited energy: the power consumption of current RF transceivers, for example, is too high for dust nodes.

- Limited computing and memory resources: many wideband ultrasound location systems, for example, sample at more than 40kHz and do signal processing on the sampled data, resulting in large memory and processing overheads [13], which is not possible on a Smart Dust node.

- Single-hop network topology: current WSN location systems often assume multi-hop networks, where a node can cooperate with its neighbors in order to compute its location [3, 25], which is likely not true in near-future Smart Dust systems.

3   Localization Challenges for Smart Dust

In this section we examine the challenges a Smart Dust system as outlined in the previous section presents for a location system.

3.1   Accuracy

The accuracy of the physical location estimates required by a Smart Dust deployment very much depends on the sensed phenomenon and the accuracy requirements of the application. In tracking applications, for example, a location grain size of about the size of the monitored phenomenon is often sufficient. That is, the required accuracy of the location system is fine-grained and ranges from centimeters (for tracking a flying insect) to meters (for tracking a large animal herd).

Note that in a typical deployment, the distance between the base station and dust nodes is in the order of tens of meters. A location system for Smart Dust should provide the desired degree of accuracy even under this condition.

3.2   Localized Location Computation

For certain applications, Smart Dust nodes need to know their own physical locations. In the following we will point out reasons for localized location computation.

In a typical application, dust nodes sample environmental parameters by reading attached sensors at regular intervals. The obtained time series of sensor readings are then preprocessed in some application-specific way before sending off relevant data to the base station.

It is a well-known observation from statistical data management that areas where changes are happening most rapidly (hot spots) should be sampled at a higher rate [11]. On the other hand, the sampling rate should be as small as possible to save energy. *Location-dependent queries* offer a solution for this tradeoff by making certain query parameters – such as the sensor sampling rate – a function of node location. A location-dependent query can be sent to the whole Smart Dust network with a single (logical) broadcast. Nodes have to obey the query parameters according to their (mutable) current location.

In order to save scarce communication bandwidth and energy, dust nodes typically cannot report detailed time series of sensor readings to the base station [11]. Instead, nodes preprocess such time series locally [28] in order to come up with a smaller and more high-level data representation (e.g., histogram or distribution function), which is then sent to the base station rather infrequently. For many applications (e.g., monitoring the spatial distribution of air pollution), preprocessing depends on the (mutable) physical location at which individual sensor readings are obtained.

In many traditional location systems such as [30], an external infrastructure observes objects and computes their location. This approach moves the burden of location computation from the nodes to a more powerful infrastructure. In Smart Dust applications where nodes need to know their location, however, the base station would have to send an individual location update message to each node of the network one by one. That is, the associated communication overhead grows linearly with the number of nodes. The following example shows that sending location updates to a network of 1000 mobile dust nodes every 20 seconds can hog all the communication bandwidth.

Sending a location update to a node involves aiming the laser beam at the node and sending a location update message. Aiming the beam typically involves aligning a steerable MEMS mirror which operates at a few hundred Hertz [21]. We will assume a mirror bandwidth of 100 Hz, a downlink communication bandwidth of 10 kbit per second, and an update message size of 20 bytes or 160 bits (3*4 bytes for physical location, plus node addressing and protocol overhead). With these parameters, the base station can send a location update to a single node about every 0.02s. Sending location updates to all nodes of a network with 1000 mobile nodes will then take 20 seconds.

Another reason for localized location computation is privacy. If dust nodes are attached to people, places or things, knowing the location of the node would also disclose the location of its host to the infrastructure. This, however, is valuable information in many cases, which can be easily abused for recording the behavior of people [20, 24]. Therefore, whenever possible, it is favorable to compute locations in the nodes themselves without disclosing them to a potentially untrusted infrastructure.

3.3   Low Cost

A location sensing system for Smart Dust imposes certain space, capital, and time costs [16]. These are due to software and hardware required for location sensing on the dust nodes and in the infrastructure (i.e., the base station). Space costs involve the amount of installed infrastructure and the node hardware's size and form factor. Capital costs include factors such as the additional price per Dust node and base station. Time costs include the overhead for system installation, calibration, and administration.

The envisioned application areas for Smart Dust impose certain limits on these costs. The intended low capital cost and small size, for example, require that the location sensing hardware overhead needed on the nodes is minimal. Ideally, a location system would reuse the existing optical receiver instead of adding additional hardware to the nodes. On the other hand, adding additional hardware to the base station is not so critical, because there will be very few base stations
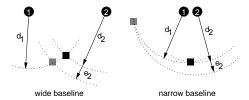
Figure 1: Errors in estimated node location depend on whether or not the points of reference for multilateration form a wide baseline.

when compared to the number of deployed dust nodes. However, introducing additional infrastructure components is not a good idea, because installation and administration of the latter contradicts the *ad hoc* nature of sensor networks.

Note that the limitation to a single piece of infrastructure (i.e., the base station) is a challenging one. In the Smart Dust single-hop network, where nodes cannot communicate directly with each other, node localization requires an external infrastructure. In multilateration-based systems, for example, the distances $d_1, ..., d_N$ to multiple points of reference $1, ..., N$ provided by the infrastructure are measured and used to compute the node's location. In order to achieve high accuracy, the reference points should form a *wide baseline*, that is, the distances among the reference points should be in the order of the distance of the node to the reference points. Figure 1 illustrates this situation in 2D. There, the distances d_1 and d_2 of the node to the two reference points 1 and 2 are measured. The node's location is computed as the intersection point of two circles with radius d_1 and d_2 centered at the reference points. If the two reference points form a wide baseline, an error e_2 in the distance measurement d_2 causes only a small error in the estimated node location. If the two reference points are close together, the same error e_2 causes a large error in the estimated node position.

Implementing a wide baseline typically requires multiple geographically distinct infrastructure components in order to provide the reference points (*beacons*). Moreover, placement of the beacons is often a non-trivial problem [5]. Usually, the exact locations of the reference points have to be known in order to compute node locations [4, 15, 24, 30]. In some systems, the beacons even need accurately synchronized clocks [18]. In order to avoid these problems, we developed a new localization approach based on cylindrical lateration, which does not have a wide baseline requirement.

Another important overhead involved in setting up a localization system is node calibration [32] in order to enforce a correct mapping of sensor readings to location estimates. In systems based on RF received signal strength (RSSI), for example, the received signal strength is mapped to a range estimate. Variations in transmit power and frequency among the nodes can cause significant inaccuracies in the range estimates when used without calibration [17]. Since the cheap
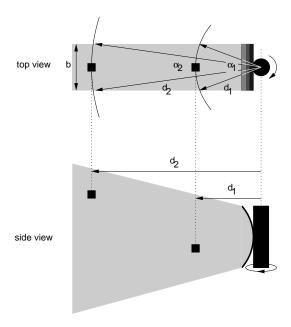

Figure 2: Top and side view of an idealistic lighthouse with a parallel beam of light.

low-power hardware used in WSN typically introduces a high variability between nodes, sensor nodes have to be individually calibrated. This, however, may not be feasible in Smart Dust installations due to their expected large scale. The Lighthouse location system does not require node calibration and thus completely eliminates the overhead of the latter.

4   The Lighthouse Location System

This section presents the Lighthouse location system for Smart Dust. In order to point out the basic ideas behind this system, we will first examine a simplified idealistic system. This examination will be followed by a more thorough discussion of a realistic system that can actually be built. We will go on by presenting a first prototype implementation, an initial set of measurements, and a first analysis of several aspects of the system.

4.1   An Idealistic System

Consider the special lighthouse depicted in Figure 2, which has the property that the emitted beam of light is parallel (i.e., has a constant width) with width b when seen from top. When seen from the side, the angle of beam spread of the parallel beam is large enough so that it can be seen from most points in space.

When this parallel beam passes by an observer, he will see the lighthouse flash for a certain period of time t_{beam}. Note that

t_{beam} depends on the observer's distance d from the rotation axis of the lighthouse since the beam is parallel. Assuming the lighthouse takes t_{turn} for a complete rotation, we can express the angle α, under which the observer sees the beam of light as follows:

$$\alpha = 2\pi \frac{t_{\mathrm{beam}}}{t_{\mathrm{turn}}} \tag{1}$$

Figure 2 shows two observers (depicted as squares) at distances d_1 and d_2 and the respective angles α_1 and α_2. Now we can express d in terms of α and the width b of the beam as follows:

$$d = \frac{b}{2\sin(\alpha/2)} \tag{2}$$

By combining Equations 1 and 2 we obtain the following formula for d in terms of b, t_{beam}, and t_{turn}:

$$d = \frac{b}{2\sin(\pi t_{\mathrm{beam}}/t_{\mathrm{turn}})} \tag{3}$$

Note that the distance d obtained this way is the distance of the observer to the lighthouse rotation axis as depicted in the side view in Figure 2. That is, all the points in space with distance d form a cylinder (not a sphere!) with radius d centered at the lighthouse rotation axis.

Based on the above observations, we can build a simple ranging system consisting of a lighthouse and an observer. The observer device contains a photo detector and a clock. When the photo detector first sees the light it records the corresponding point in time t_1. When the photo detector no longer sees the light it records t_2. When it sees the light again it records t_3. With $t_{\mathrm{beam}} := t_2 - t_1$ and $t_{\mathrm{turn}} := t_3 - t_1$ the observer can apply Equation 3 in order to calculate its distance d from the lighthouse rotation axis. Note that if t_{turn} is constant it has to be measured only once since it does not change with distance. Also note that the necessary hardware resources of the observer device are matched by a Smart Dust node as explained in Section 2.

This ranging scheme can be used to build a *single device*, which allows observers to autonomously determine their position relative to it in three dimensional space. This device consist of three lighthouses with mutually perpendicular rotation axes as depicted in Figure 3. Assuming an observer measures the distances d_x, d_y, and d_z as indicated above, its location can be determined by computing the intersection point(s) of three cylinders with radius d_x, d_y, d_z centered at the respective lighthouse rotation axes. Note that there are 8 such intersection points in general, one in each of the 8 quadrants of the coordinate system. If we can ensure, however, that all observers are located in a single quadrant (e.g., the main quadrant defined by the points (h_x, h_y, h_z) with $h_x, h_y, h_z \geq 0$), there is a unique intersection point. This
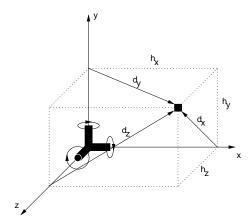

Figure 3: 3D Localization support device consisting of three mutually perpendicular lighthouses.

intersection point can be obtained by solving the following equation system for h_x, h_y, h_z:

$$\begin{array}{rcl} d_x^2 & = & h_y^2 + h_z^2 \\ d_y^2 & = & h_x^2 + h_z^2 \\ d_z^2 & = & h_x^2 + h_y^2 \end{array} \tag{4}$$

Note that this equation system does not necessarily have a solution, since the values d_x, d_y, d_z are only approximations obtained by measurements. If there is no solution, an approximation for the intersection point can be obtained using minimum mean square error (MMSE) methods. The solution (h_x, h_y, h_z) obtained this way minimizes the sum of the squares of the differences of the left hand and right hand sides of the equations 4. However, if the equation system has a solution, it can be directly solved using the following set of equations, again assuming that the observer is located in the main quadrant of the coordinate system depicted in Figure 3:

$$\begin{array}{rcl} h_x & = & \sqrt{(-d_x^2 + d_y^2 + d_z^2)/2} \\ h_y & = & \sqrt{(d_x^2 - d_y^2 + d_z^2)/2} \\ h_z & = & \sqrt{(d_x^2 + d_y^2 - d_z^2)/2} \end{array} \tag{5}$$

The setup of the complete location system can now be described. The base station is equipped with three mutually perpendicular lighthouses as depicted in Figure 3. At startup, the base station broadcasts certain calibration parameters (e.g., the beam width b for each of the lighthouses) to all dust nodes. The latter use a real-time clock to measure the amount of time during which each of the lighthouses beams are visible. Using Equations 3 and 4, nodes can autonomously compute their location in the reference grid defined by the base station's three lighthouses.

The description of the system's principles gives rise to a number of practical questions. First of all, it is not clear at all
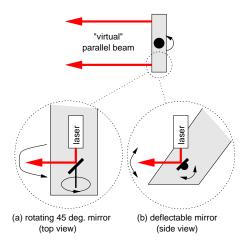
Figure 4: A rotating lighthouse with a "virtual" parallel beam whose outline is defined by two parallel laser beams. Rotating (a) or deflectable mirrors (b) are used make the laser beams scan the northern hemisphere of the lighthouse.

whether a system fulfilling the above requirements (e.g., parallel beam) can actually be built in practice. Moreover, we did not discuss the problem how a dust node can distinguish the different beams of the lighthouses, or what happens if a dust node "sees" the beams of two lighthouses at the same time. We will discuss these issues in the next sections in order to lay the foundation for an implementation of the system.

4.2 A Realistic System

During the first experiments it turned out that actually building a lighthouse with a sufficiently exact parallel beam is very difficult, at least given the limited technical capabilities that were available to us. This has the unfortunate consequence, that the model described in Section 4.1 cannot directly be used due to the resulting high inaccuracies. To understand the reason of these inaccuracies, consider the following example, where we assume a beam width of 10cm. Even if the angle of beam spread is only $1°$ (instead of $0°$ for an ideal parallel beam), the width of the beam at a distance of 5m would be about 18.7cm, resulting in an error of almost 90%. The relative error could be reduced somewhat by increasing the width of the beam. However, a large beam width also results in a large and clumsy base station device.

Therefore, instead of building a system perfectly matching the requirements of Section 4.1, we have to adapt our model to a system which can actually be built. In order to develop such a model, we first have to examine ways of generating near-parallel beams.

4.2.1 Beam Generation

In order to keep the hardware and energy overhead on the Smart Dust nodes small, the beam must be easily detectable. Furthermore, the system should work with high accuracy even if the base station is far away (tens of meters, say) from the nodes. Therefore we decided to use a laser-based approach. As mentioned above, the beam should be as wide as possible in order to keep inaccuracies small. In order to achieve this, we use *two* lasers to create the *outline* of a parallel beam as depicted in the upper half of Figure 4. This makes no difference to a single wide beam, since we are only interested in the edges of the beam (i.e., change from "dark" to "light" and vice versa) in order to measure t_{beam} and t_{turn}.

Due to the narrow laser beams, the "virtual" parallel beam generated this way can only be seen from a single plane, however. In order to ensure that the beam can be seen from any point in the northern hemisphere of the lighthouse without defocusing the lasers, the laser beams have to scan this space in some way. The lower half of Figure 4 depicts two ways to achieve this. The first approach uses a small mirror mounted on a rotating axle under an angle of $45°$. By pointing the laser at this mirror, the reflected rotating beam describes a plane. With commercial off the shelf technology we can easily achieve a rotation frequency of about 300Hz. The second approach uses a small deflectable MEMS mirror similar to the one used as part of the corner cube retroreflector (CCR). The MEMS mirror presented in [7], for example, operates at 35kHz and achieves a deflection angle of $25°$. A laser beam pointed at such a mirror can thus sweep over an angle of $50°$ at a frequency of 35kHz.

Based on this approach, a lighthouse consists of a (slowly) rotating platform, on which two semiconductor laser modules and two rotating (or deflectable) mirrors are mounted. However, as mentioned at the beginning of Section 4.2, it is next to impossible to assemble all the pieces such that the resulting "virtual" wide beam is almost parallel. Therefore, we have to come up with a model which describes an imperfect but realistic system. The model discussed below is based on rotating mirrors, since we used this approach in our prototype implementation of the system. However, the model equally applies to a system based on deflectable mirrors.

4.2.2 The Lighthouse Model

We use Figure 5 to explain the lighthouse model. It shows a simplified top and side view of the lighthouse. Each view shows the two mirror's rotation axes and the corresponding reflected rotating laser beams. Note that in general the angle enclosed by the mirror rotation axis and the mirror will not be exactly $45°$ (i.e., $\beta_i \neq 0°$) due to manufacturing limitations. Therefore, the rotating reflected laser beams will form two cones as depicted in Figure 5. Moreover, the two mirror's
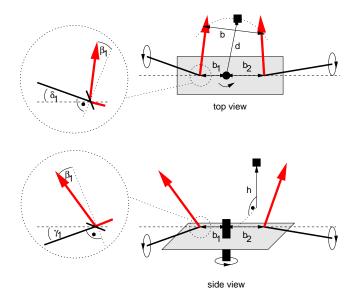
Figure 5: Model of a realistic lighthouse based on rotating mirrors. The zoom-ins show detail for one rotating mirror in the top and side views. The other rotating mirror has respective parameters β_2, γ_2, and δ_2.

rotation axes will not be perfectly aligned. Instead, the dashed vertical line (connecting the apexes of the two cones formed by the rotating laser beams) and the mirror rotation axes will enclose angles γ_i in the side view and angles δ_i in the top view that are different from $0°$. Additionally, the figure shows the rotation axis of the lighthouse platform and its distances b_1 and b_2 to the apexes of the two light cones. The *lighthouse center* is defined as the intersection point of the lighthouse platform rotation axis and the dashed vertical line in Figure 5. Note that the idealistic lighthouse described in Section 4.1 is a special case of this more complex model with $\beta_i = \gamma_i = \delta_i = 0°$ and $b_1 = b_2$.

Now let us consider an observer (black square) located at distance d from the main lighthouse platform rotation axis and at height h over the lighthouse center. We are interested in the width b of the virtual wide beam as seen by the observer. Let us assume for this that we can build a lighthouse with $b_1 \approx b_2$ and $\beta_i, \gamma_i, \delta_i \approx 0°$, i.e., we do our best to approximate the perfect lighthouse described in Section 4.1. Then we can express b approximately as follows:

$$\begin{aligned} b &\approx & b_1 + b_2 + \sqrt{d^2 + h^2}(\sin\beta_1 + \sin\beta_2) + \\ & & h(\tan\gamma_1 + \tan\gamma_2) + d(\sin\delta_1 + \sin\delta_2) \end{aligned} \quad (6)$$

The inaccuracy results from the last two terms, which are linear approximations of rather complex non-linear expressions. For $\beta_1 = \beta_2 = 0°$, however, expression 6 becomes an equation. We will allow these factors to be built into the error term.

With $C^b := b_1 + b_2$, $C^\beta := \sin\beta_1 + \sin\beta_2$, $C^\gamma := \tan\gamma_1 + \tan\gamma_2$, and $C^\delta := \sin\delta_1 + \sin\delta_2$ we can rewrite expression 6 as

$$b \approx C^b + \sqrt{d^2 + h^2}C^\beta + hC^\gamma + dC^\delta \quad (7)$$

Note that C^b, C^β, C^γ, and C^δ are fixed lighthouse parameters. We will show below how they can be determined using a simple calibration procedure. We can express b also in terms of the angle α obtained using Equation 2:

$$b = 2d\sin\frac{\alpha}{2} \quad (8)$$

Combining expressions 7 and 8 we obtain the following expression which defines the possible (d, h) locations of the observer given a measured angle α and the lighthouse calibration values C^*:

$$2d\sin\frac{\alpha}{2} \approx C^b + \sqrt{d^2 + h^2}C^\beta + hC^\gamma + dC^\delta \quad (9)$$

Note that for given C^* and α the points in space whose d and h values are solutions of Equation 9 form a rotational hyperboloid centered at the rotation axis of the lighthouse. In the special case $\beta_i = \gamma_i = \delta_i = 0°$ and $b_1 = b_2$ this hyperboloid becomes a cylinder as in the idealistic model described in Section 4.1.

4.2.3 Location Computation

Similar to the idealistic model described in Section 4.1, the location of the observer can be obtained by determining the intersection point(s) of the three rotational hyperboloids defined by Equation 9. However, since the observed virtual beam width b now additionally depends on the height h of the observer, we have to take into account the exact lighthouse positions. Figure 6, which shows an extended version of Figure 3, illustrates this. The marks on the coordinate axes show the positions of the lighthouse center (as defined in Section 4.2.2) of each of the three lighthouses. That is, the coordinates of the observer are $(x_0 + h_x, y_0 + h_y, z_0 + h_z)$ with respect to the origin formed by the intersection of the three lighthouse rotation axes. In order to obtain approximations for the values h_x, h_y, and h_z, we have to solve the following equation system:

$$\begin{aligned} 2d_x\sin\frac{\alpha_x}{2} &= C_x^b + \sqrt{d_x^2 + h_x^2}C_x^\beta + h_xC_x^\gamma + d_xC_x^\delta \\ 2d_y\sin\frac{\alpha_y}{2} &= C_y^b + \sqrt{d_y^2 + h_y^2}C_y^\beta + h_yC_y^\gamma + d_yC_y^\delta \\ 2d_z\sin\frac{\alpha_z}{2} &= C_z^b + \sqrt{d_z^2 + h_z^2}C_z^\beta + h_zC_z^\gamma + d_zC_z^\delta \\ d_x^2 &= (y_0 + h_y)^2 + (z_0 + h_z)^2 \\ d_y^2 &= (x_0 + h_x)^2 + (z_0 + h_z)^2 \\ d_z^2 &= (x_0 + h_x)^2 + (y_0 + h_y)^2 \end{aligned}$$
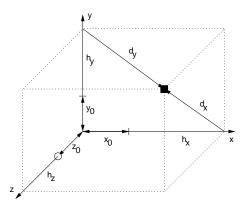
$$(10)$$

Figure 6: Positions of the lighthouses in the coordinate system. d_z is not shown for clarity.

The indices $\{x, y, z\}$ indicate which lighthouse the values are associated with. As with equation system 4, this system does not necessarily have a solution, since the parameters are only approximations obtained by measurements. Therefore, minimum mean square error (MMSE) methods have to be used to obtain approximations for the h_*. However, if the equation system 10 has a solution, we can approximately solve it by simple iteration. For this, we first transform each of the six equations of equation system 10 in order to obtain the following fixpoint form:

$$
\begin{aligned}
h_x &= f_1(d_x) \\
h_y &= f_2(d_y) \\
h_z &= f_3(d_z) \\
d_x &= f_4(h_y, h_z) \\
d_y &= f_5(h_x, h_z) \\
d_z &= f_6(h_x, h_y)
\end{aligned}
\tag{11}
$$

Note that we did not show arguments of the f_i (i.e., C_*^*, α_*, x_0, y_0, z_0) that do not change during iterative evaluation of the equation system. By using appropriate values for h_x^0, h_y^0, h_z^0, and Δ, we can obtain approximate solutions for h_x, h_y, h_z with the following algorithm:

```
h_x := h_x^0;
h_y := h_y^0;
h_z := h_z^0;
while (true) {
        h_x' := f_1(f_4(h_y, h_z));
        h_y' := f_2(f_5(h_x, h_z));
        h_z' := f_3(f_6(h_x, h_y));
        if (|h_x' - h_x| + |h_y' - h_y| + |h_z' - h_z| < Δ)
                break;
        h_x := h_x';
        h_y := h_y';
        h_z := h_z';
}
```

At first, the h_* are initialized to the start values h_*^0. Using the f_i, new approximations h_*' are computed. We are finished if the new values are reasonably close to the original h_*. Otherwise we update the h_* to the new values and do another iteration. For good convergence of this algorithm the partial derivatives of the $f_i \circ f_{3+i}$ in the environment of the solution (h_x, h_y, h_z) should be small, which is typically true. In our prototype implementation we use $h_*^0 := 100\text{cm}$ and $\Delta := 0.1\text{cm}$. With this configuration, the algorithm typically performs 4-6 iterations.

4.2.4 Calibration

What remains to be shown is how we can obtain values for x_0, y_0, z_0, and C_x^*, C_y^*, C_z^*. Since the values x_0, y_0, z_0 are uncritical for the achieved accuracy, we assume they are measured directly. The C_*^* values, however, are very critical for the accuracy as was shown with the example at the beginning of Section 4.2. Therefore we have to perform a calibration.

For each of the three lighthouses we have to determine values for the four variables $C^b, C^\beta, C^\gamma, C^\delta$. For this, we place the observer at known locations (d_i, h_i) and obtain the respective α_i using equation 1. Doing so for at least four locations and using equation 9, we obtain the following linear equation system in $C^b, C^\beta, C^\gamma, C^\delta$:

$$
\begin{aligned}
2d_1 \sin \tfrac{\alpha_1}{2} &= C^b + \sqrt{d_1^2 + h_1^2}\,C^\beta + h_1 C^\gamma + d_1 C^\delta \\
2d_2 \sin \tfrac{\alpha_2}{2} &= C^b + \sqrt{d_2^2 + h_2^2}\,C^\beta + h_2 C^\gamma + d_2 C^\delta \\
2d_3 \sin \tfrac{\alpha_3}{2} &= C^b + \sqrt{d_3^2 + h_3^2}\,C^\beta + h_3 C^\gamma + d_3 C^\delta \\
2d_4 \sin \tfrac{\alpha_4}{2} &= C^b + \sqrt{d_4^2 + h_4^2}\,C^\beta + h_4 C^\gamma + d_4 C^\delta
\end{aligned}
\tag{12}
$$

As with the other equation systems, this system does not necessarily have a solution, since the parameters are only approximations obtained by measurements. Again, MMSE methods can be used to obtain approximations for the C^*. If the system has a solution, it can also be obtained by Gaussian elimination. For this, the d_i and h_i have to fulfill certain requirements. One simple rule of thumb is that both the d_i and the h_i should be pairwise distinct.

Note that calibration has to be performed only once for each base station (assuming that the system is stable enough and needs not be recalibrated) and is independent of the receiver nodes. Therefore, calibration can be performed using a more powerful receiver device than the limited Smart Dust node. As explained in Section 4.1, the base station broadcasts these calibration parameters to the Smart Dust nodes, which use them to compute their location using Equation System 10.
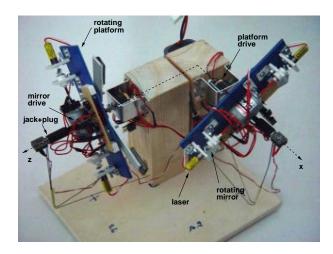
Figure 7: Prototype base station consisting of two lighthouses and the resulting 2D coordinate system.

4.3 Prototype Implementation

In order to evaluate the concepts developed in Section 4.2, we implemented a first prototype system. To keep the hardware overhead small, this prototype system consists of only two lighthouses and allows observers located on the plane $y = 0$ to determine their x and z coordinates. From a conceptual point of view, the differences to a 3D system are minimal.

4.3.1 The Base Station

Figure 7 shows a picture of the prototype base station. It consists of two mutually perpendicular lighthouses. The main lighthouse platform takes about $t_{\text{turn}} = 60$s for one rotation. The platform is driven by a geared electro motor manufactured by FTB [34], which has a low flutter of about 0.1% of the rotation speed. Using an LM317 [36] adjustable voltage regulator, the voltage supply of the motor and thus the rotation speed of the platform can be adjusted. The two bars that extend from under the platform are used to move the center of gravity of the platform to the rotation axis, such that the platform rotates at a constant speed.

The power supply for the rotating platform is implemented by a stereo jack and associated plug. While the plug is fixed to the axle of the rotating platform, the jack is affixed to the chassis using a thin steel wire. This way, the round plug can rotate in the jack.

Beam generation is based on rotating mirrors as described in Section 4.2. Both rotating mirrors are driven by a single Graupner SPEED 280 electro-motor. In order to reduce vibrations, we did not use a rigid axle to connect the mirrors to the motor. Instead, we used small steel springs as axles. The rotating mirrors are supported by two ball bearings each. Two
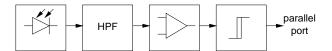

Figure 8: Schematic diagram of the receiver hardware.

1mW 650nm semiconductor laser modules with adjustable focus point their beam at the rotating mirrors.

The supply voltage of the motor and thus its rotation speed can be adjusted using an LM317 voltage regulator. The mirror rotation speeds of the two lighthouses are slightly different ($t_{\text{mirror}} = 4$ms and $t_{\text{mirror}} = 5$ms for one rotation, respectively), such that the observer can distinguish the two lighthouses based on the time interval between successive light flashes, which will be explained in more detail in Section 4.3.2. Hence, in order to detect a beam, the observer's photo detector must at least be hit twice by the rotating laser beam. Note that due to the fast rotation of the laser beams, the average light intensity is low enough to be eye-safe.

There is a slight chance that the photo detector is hit by the beams of both lighthouses at the same time. We will explain in Section 4.3.2 how an observer can detect and handle this situation. However, since the diameter of the laser beams is rather small, the likelihood of this event is small. By selecting slightly different platform rotation speeds for the two lighthouses, we can ensure that for each observer this happens only once in a while. In our experiments this happened about every 100 lighthouse rotations at a single fixed observer.

4.3.2 The Nodes

The receiver prototype consists of a small electronic circuit connected to the parallel port of a laptop computer running Linux. Figure 8 shows a schematic diagram of the receiver hardware. A photo diode converts the intensity of the incident light into a proportional voltage. The light that is incident to the photo diode mainly consists of three components:

- direct current (DC) components resulting from slowly changing daylight

- low frequency components resulting from artificial lighting powered with 50Hz alternating current (AC)

- higher frequency components resulting from laser light flashes at about 200Hz-300Hz ($1/t_{\text{mirror}}$)

Since we are only interested in the higher frequency laser flashes, we run the output signal of the photo diode through a high pass filter (HPF) which removes DC and low frequency components. Due to this, the detector is insensitive to daylight and artificial light.
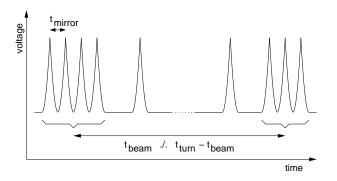
Figure 9: Input voltage at the parallel port as beams pass by the photo detector.

The output of the HPF is then amplified using an operational amplifier, whose output is in turn fed into a Schmitt Trigger. The latter implements a hysteresis, i.e., when the input voltage level exceeds a certain value V_1 it lowers the output voltage to a minimum. When the input voltage falls below a certain value V_2, the Schmitt Trigger raises the output voltage to a maximum. The output of the Schmitt Trigger is connected to the parallel port, so that each laser flash on the photo diode causes a parallel port interrupt to be triggered.

The receiver software consists of two main components, a Linux device driver which handles the parallel port interrupt, and an application level program which performs the actual location computation and lighthouse calibration. The device driver mainly consists of the parallel port interrupt handler, which is implemented using the parapin [37] parallel port programming library. Moreover, it implements a Linux special device /proc/location, which provides a simple interface to user level applications. By writing simple ASCII commands to this device, a user level program can instruct the device driver to do some action. By reading the /proc/location device, a user level program can obtain the current status and measured angle α according to Equation 1 of all detected lighthouses.

In order to measure α, the driver has to evaluate the interrupts it sees. To understand how this is done, consider Figure 9, which shows the input voltage at the parallel port over time. As the first rotating laser beam passes by the photo detector, the parallel port sees a sequence of sharp pulses resulting from the fast rotating mirror. The pulses stop if the lighthouse platform has turned enough so that the photo detector isn't hit any longer by the rotating beam. After some time, the second rotating beam passes by the photo detector and again generates a sequence of fast pulses.

Recall that each pulse generates an interrupt, which results in the device driver interrupt handler being invoked. The handler then uses the system clock (which has μs resolution under Linux) to determine the point in time when the interrupt occurred.

The time interval between two successive fast pulses equals the time t_{mirror} for one rotation of the mirror. Since each lighthouse has a different t_{mirror}, this value can be used to distinguish different lighthouses. Please note that the pulse sequences can contain "holes" where the laser beam missed the photo detector due to vibrations. The driver removes all peaks separated by holes from the beginning and the end of the sequence of pulses. The time median of the resulting shorter sequence of pulses without holes is assumed as the detection time of the beam (indicated by the braces in Figure 9).

Recall from Section 4.2, that we implemented a "virtual" wide beam by two rotating laser beams that form the outline of this wide beam. Therefore, the time passed between the medians of two successive packs is either t_{beam} or $t_{\mathrm{turn}} - t_{\mathrm{beam}}$. If the actual value is small (e.g., $< 1\mathrm{sec}$) then it is assumed to be t_{turn}. If the lighthouse has just been initialized the driver also measures $t_{\mathrm{turn}} - t_{\mathrm{beam}}$ in order to obtain t_{turn}. Since the latter does not change, this has to be done only once. Later on, the driver can output a new α with each round of the lighthouse.

In order to distinguish successive pulses from "holes", and holes from "beam switches", the driver knows tight lower and upper bounds for the possible values of t_{mirror} and t_{turn}. In Section 4.3.1 we mentioned the possibility that beams from different lighthouses may hit the photo receiver at the same time. If this happens the resulting time between successive pulse will fall below the lower bound for t_{mirror}, such that the driver can detect this situation instead of producing faulty results.

We also ported the receiver hardware and software to an AT-MEL AT128L 8-bit embedded micro controller [38]. This setup more closely resembles the limited capabilities of a Smart Dust node and allows us to study the potential effects of a Smart Dust node on the location system.

4.4   Measurements

In this section we present an initial benchmark obtained with the prototype described above. We will begin by describing the calibration procedure.

4.4.1   Calibration

Calibration of the base station involves the following three steps:

- Ensuring that the lighthouses are mutually perpendicular.

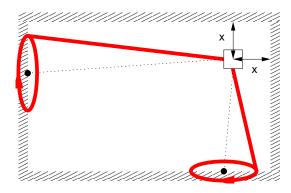- Measuring the offsets of the lighthouse centers x_0 and z_0.

Figure 10: Ensuring mutually perpendicular lighthouses.

- Determining $C^b, C^\beta, C^\gamma, C^\delta$ for each of the two lighthouses.

In order to ensure that the two lighthouses are mutually perpendicular, we placed the base station at the corner of a rectangular room as depicted in Figure 10, such that the rotation axes of the two lighthouses are at distance x from the two perpendicular walls. We disabled the motors that drive the rotating mirrors and one of the two lasers of each lighthouse. Then we adjusted the mirror so that the remaining laser beam points at the opposite wall. Due to the rotating lighthouse platforms, the laser spots draw two circles on the walls. The centers of these two circles mark the position where the lighthouse rotation axes hit the wall as depicted in Figure 10. Now we adjust the lighthouses on the common chassis such that the centers of these circles also have a distance x from the walls. In our measurement setup, we placed the base station at $x = 20cm$ in a room with a size of about 5m by 5m.

As mentioned in Section 4.2, the lighthouse center offsets x_0 and z_0 from the origin of the coordinate system defined by the lighthouse rotation axes are not critical for the accuracy of the system. Therefore, we measured them directly at the base station device.

In order to determine the C^* values, we placed the observer at the four locations $(x, z) \in \{(50, 50), (480, 80), (80, 480), (340, 340)\}$ (all values in centimeters) on the floor in the base station coordinate system. The respective lighthouse distance and height values are obtained from the (x, z) values as follows:

$$
\begin{aligned}
(d_x, h_x) &:= (z, x - x_0) \\
(d_z, h_z) &:= (x, z - z_0)
\end{aligned}
\tag{13}
$$

At each of the locations we performed ten measurements of α for each lighthouse and computed the mean value. For each of the two lighthouses we then solved Equation System 12 in order to obtain the C^* values.
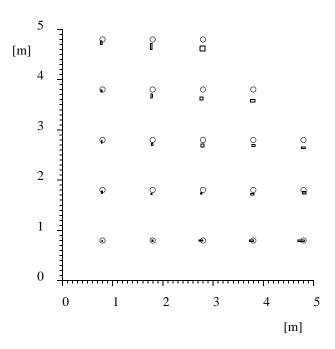

Figure 11: Location estimation benchmark. The ground truth locations are at the centers of the circles. The mean of the measured locations are at the centers of the boxes. The edge length of each box is twice the standard deviation in each axis.

4.4.2 Benchmark

For the benchmark, we placed the observer at 22 locations on the grid $(80cm + i * 100cm, 80cm + j * 100cm)$ in the base station coordinate system on the floor of the room. At each of the locations, we measured the location ten times by iteratively solving Equation System 10 as described in Section 4.2.

Figure 11 shows the base station coordinate system and the results of these measurements. Ground truth locations (x, z) are indicated by circles. The mean of the computed location $(\bar{x}, \bar{z})$ is at the center of the small boxes. The edge length of each box is twice the standard deviation s_x (s_z) of the measurements in the respective axis.

Please note that we determined ground truth locations using a cheap 5m tape measure, resulting in a maximum error of about $\pm 1cm$ in each axis. Also note that we did not perform out-lier rejection or any other statistical "tricks" to improve the mean values or standard deviations.

The mean relative offset of the mean locations from ground truth locations (i.e., $|\bar{x} - x|/x$) is 1.1% in the x axis, and 2.8% in the z axis. The overall mean relative offset of the mean locations from ground truth locations (i.e., $|(\bar{x}, \bar{z}) - (x, z)|/|(x, z)|$ is 2.2%. The mean relative standard deviation (i.e., s_x/x) is 0.71% in the x axis and 0.74% in the z axis. The overall mean relative standard deviation (i.e., $s_{|(x,z)|}/|(x, z)|$) is 0.68%.

Note that while the mean standard deviations are almost the same for both axes, the mean relative offset of 2.8% in the z axis is more than twice the value for the x axis. We believe that this is due to the way we performed calibration. Firstly, some of the locations where we performed measurements are outside of the convex hull of the locations where we performed calibration. Additionally, we calibrated at only four locations and solved Equation System 12 directly in order to obtain the C^* values. We expect better results by performing calibration using a larger set of reference locations and by using MMSE methods as mentioned in Section 4.2. We are currently working on improving the calibration part of our software.

4.5 System Analysis

This section presents a first analysis of several aspects of the Lighthouse location system, namely factors that influence the accuracy of the location estimates, limits on the maximum distance of nodes from the base station, the effects of node mobility, and the cost of adding location support to dust nodes.

4.5.1 Accuracy

In this section we want to examine which factors influence the accuracy of the system. For this, we have to examine errors that can occur during the measurement of t_{beam} and t_{turn}. From a measurement point of view the two are identical, since they are both an amount of time elapsed between two beam sightings. Therefore we will use t as a genus for the two and Δt as the absolute error of t. The following list contains possible causes for measurement errors:

- Vibrations: Due to their fast rotation, the mirrors and thus the reflected beams suffer from small vibrations, resulting in a small angle ϵ of beam spread, which is about $0.05°$ in our prototype. Assuming $\sin \epsilon = \epsilon$ (since $\epsilon \approx 0$), the resulting error is $\Delta t \leq t_{\text{turn}} \frac{\epsilon \sqrt{d^2 + h^2}}{2\pi d}$ for an observer located at distance d from the lighthouse rotation axis and at height h over the lighthouse center.

- Lower bound on time t_{mirror} for one mirror rotation: Since we can measure elapsed time only when the rotating laser beam hits the photo detector, the accuracy of t_{beam} and t_{turn} is limited by the speed of the rotating mirrors (i.e., t_{mirror}). The resulting error is $\Delta t \leq t_{\text{mirror}}$.

- Flutter of platform rotation: The relative error in lighthouse rotation speed ρ_{lh} causes an error in t. ρ_{lh} is mainly caused by the flutter of the motor driving the lighthouse platform. The motor used in our prototype has a flutter of 0.1%. The resulting error is $\Delta t \leq t\rho_{\text{lh}}$.
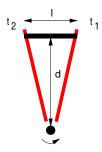

Figure 12: The photo detector must be hit by the laser beam at least twice.

- Variable delays: There is a variable time offset between the laser beam hitting the photo detector and the interrupt handler reading the clock. On the path from the photo detector to the interrupt handler are many sources of variable delay, such as hardware and interrupt latency. The actual value of this error pretty much depends on what is currently happening on the computer, but is typically small compared to the other sources of errors.

- Clock resolution: The minimum time unit t_{min} that can be measured by the clock limits the time resolution for measurement of t. The Linux laptop we used has $t_{\text{clockres}} = 1\mu s$. On the ATMEL we used a 16-bit counter to implement a clock with $t_{\text{clockres}} = 50\mu s$. The resulting error is $\Delta t \leq t_{\text{clockres}}$.

- Clock drift: The maximum relative error ρ_{clock} in the clock rate also causes an error in t. A typical value is $\rho_{\text{clock}} = 10^{-5}$ both on Linux and the ATMEL. The resulting error is $\Delta t \leq t\rho_{\text{clock}}$.

In our prototype systems, the clearly dominating errors are caused by vibrations, limited t_{mirror}, and flutter of platform rotation. The use of deflectable MEMS mirrors can both drastically reduce vibrations and t_{mirror}. The flutter of platform rotation can be reduced to about 0.01% by using electronically stabilized motors as used, for example, in turntable drives. By this, we expect a possible reduction of Δt by a factor of about 10.

Note, however, that the errors resulting from these three main sources can be modeled by a Gaussian noise source. This means that averaging over a large number of measurements helps to reduce the error.

4.5.2 Range

In this section we want to examine the maximum range, at which observers can still determine their location. This maximum range mainly depends on two issues.

The first of these issues is that the photo receiver has to be hit twice by each of the rotating beams in order for the receiver to

identify the lighthouse as explained in Section 4.3.2. Figure 12 depicts this situation. It shows a top view of a lighthouse with only *one* of the two rotating beams at *two* points in time t_1 and t_2. At t_1, the beam hits the photo detector at distance d from the lighthouse rotation axis the first time. Then, the mirror does one rotation and hits the photo detector a second time at t_2. During $t_2 - t_1$, the lighthouse platform has rotated a bit to the left. l denotes the diameter of the photo detector. Assuming a constant diameter w of the laser beam, the distance d at which the photo detector is hit at least twice is given by the following inequality:

$$d < \frac{l + w}{2 \sin(\pi t_{\mathrm{mirror}} / t_{\mathrm{turn}})} \qquad (14)$$

With the values of our prototype system $l = 5\text{mm}$, $w = 3\text{mm}$, $t_{\mathrm{mirror}} = 4\text{ms}$, $t_{\mathrm{turn}} = 60\text{sec}$ we can achieve a theoretical maximum range of about 14m. This value can be improved by increasing t_{turn}, by decreasing t_{mirror}, or by defocusing the lasers a bit, such that there is a small angle of beam spread. However, there are certain limits for each of these possibilities. The angle of beam spread is limited by the sensitivity of the photo detector and the output power of the laser. t_{mirror} is limited by the possible maximum speed of the mirrors. With MEMS deflectable mirrors such as the one presented in [7], we can achieve $t_{\mathrm{mirror}} = 1/35kHz = 30\mu s$. t_{turn} is limited by the frequency of location updates needed by the nodes and thus by the degree of node mobility (see Section 4.5.4).

The second issue that limits the maximum range of the system is the speed of the photo detector. Using COTS technology, the beam has to stay on the photo detector for about $t_{\mathrm{photo}} = 10\text{ns}$ in order to be detected. Depending on the minimum retention period t_{photo} of the laser beam on the photo detector, the maximum distance d is limited according to the following inequality:

$$d < \frac{l + w}{2 \sin(\pi t_{\mathrm{photo}} / t_{\mathrm{mirror}})} \qquad (15)$$

With the current values of our prototype $t_{\mathrm{photo}} = 200\text{ns}$, $t_{\mathrm{mirror}} = 5\text{ms}$, $l = 4\text{mm}$, $w = 3\text{mm}$ we can achieve a theoretical maximum range of about 27m, giving us an overall range limit of 14m. Again, this value can be improved by reducing t_{mirror} and by defocusing the laser with the same limits as above.

The actually measured maximum range, at which the receiver prototype could still detect the base station is about 11 meters. However, the range can be increased by adjusting certain system parameters. A more elaborate system built using fast deflectable MEMS mirrors with values $l = 1\text{mm}$, $w = 20\text{mm}$ (due to beam spread), $t_{\mathrm{mirror}} = 1\text{ms}$, $t_{\mathrm{turn}} = 60\text{sec}$, and $t_{\mathrm{photo}} = 10\text{ns}$, for example, could achieve a theoretical maximum range of about 210m (the minimum obtained from Inequalities 14 and 15). Based on our experience, we would expect a practical maximum range of about 120-140m of a system with these parameters, which approximately equals the maximum communication range of 150m during the day for the Berkeley experiments [19].

4.5.3 Cost

In this section we want to examine how the presented location system fits the stringent resource restrictions of future Smart Dust Systems. As explained in Section 3.3, these restrictions especially apply to the receiver side, i.e., the Smart Dust nodes.

The Berkeley Smart Dust prototype has already demonstrated that a photo detector similar to the one we are using for our location system is feasible. What remains to be shown is how the receiver software (i.e., the device driver and the user level program) fit onto a Smart Dust node.

Both the processing overhead and the memory footprint of the device driver are very low, which is very important for Smart Dust. The first is true because the driver is interrupt driven, i.e., it does not do any sensor sampling or polling. Moreover, the interrupt can be used to wake up the processor from a power saving mode. Thus, the system has to be woken up only during the short periods when a beam is hitting the photo detector. The memory footprint is very low because the driver does not have to store arrays of peak detections. Instead, for each sequence of peaks it only keeps "first peak" and "last peak" time stamps which are updated when a new interrupt occurs. The whole data structure for one lighthouse only takes about 25 bytes.

Similarly, the location computation part of the user level program has a very low memory footprint. It just retrieves the α values from the device driver and executes the approximation program described in Section 4.2. Given the relatively infrequent location updates, speed is not a problem. On computationally very limited platforms like future Smart Dust nodes, it might be necessary to revert to fixed point arithmetic and a hardware implementation of the location computation code in case the provided processing capabilities are too limited. Besides the basic arithmetic operations $(+, -, *, /)$ we need support for $\sin \alpha$ and $\sqrt{x}$ in order to solve Equation System 10. Note that $\sin \alpha$ is easy to approximate since the values of α obtained from Equation 1 are small due to $t_{\mathrm{beam}} \ll t_{\mathrm{turn}}$. The second order approximation $\sin \alpha \approx \alpha - \alpha^3/6$ has a maximum error of 0.1% for $|\alpha| \leq 33°$. There are also fast approximations for $y = \sqrt{x}$. One possible approach is to first approximate $1/\sqrt{x}$ by iterating $y := y(3 - xy^2)/2$ with an appropriate initial value for y. Multiplying the result by x gives an approximation for $\sqrt{x}$.

The requirements on the clock are also quite relaxed. Note that we don't need a real-time clock since we are only interested in the quotient $t_{\mathrm{turn}} / t_{\mathrm{beam}}$. A simple counter which

ticks at a constant rate would also be sufficient. The resolution of the clock (or counter) just has to be high enough to reliably distinguish the t_{mirror} values of different lighthouses. Since the t_{mirror} values of our prototype system are 4ms and 5ms, respectively, a clock resolution of 0.5ms would be sufficient.

Please note that dust nodes don't have to be calibrated due to the following two reasons. Firstly, the two beam sightings used to measure t_{beam} and t_{turn} are identical from a measurement point of view. Any constant hardware and software delays will subtract out. Secondly, only the quotient $t_{\mathrm{turn}}/t_{\mathrm{beam}}$ is used for node localization, which is independent of the actual clock frequency.

4.5.4 Node Mobility

If nodes change their location over time, they have to update their location estimates frequently in order to avoid inaccuracies resulting from using outdated location estimates. Moreover, node movement during the measurement of parameters needed for location computation can cause inaccuracies in the estimated location.

The time t_{update} between successive location updates usually equals the time t_{turn} required for one rotation of the lighthouse. Thus, the update frequency $1/t_{\mathrm{update}}$ can be increased by decreasing t_{turn}. However, there is an easy way to double the update frequency when using rotating mirrors for beam generation, because the beams are reflected to both sides of the lighthouse as depicted by the dashed laser beams in Figure 5. Thus, we actually have two "virtual" wide beams we can use for location estimation, effectively doubling the update frequency.

If a node moves during measurement of t_{beam} (i.e., after detection of the first beam and before detection of the second beam), the obtained value of t_{beam} will be incorrect. Additional errors are caused by the node moving between measurements of t_{beam} of the three lighthouses.

There are two ways to detect and reject faulty location estimates resulting from node movement during measurement. The first compares two or more consecutive position estimates and rejects them if they differ by more than a small threshold. The second approach uses accelerometers to detect movement during measurement. Accelerometers can also be used to estimate node movement (velocity, direction) during measurements of t_{beam}. The obtained values can be used to correct t_{beam}, such that correct location estimates can also be obtained during node movement. In fact, the Smart Dust prototypes developed at Berkeley already contain such sensors.

4.5.5 Line-Of-Sight Requirement

As mentioned in Section 2, communication between a node and the base station requires an uninterrupted line-of-sight (LOS) even for "plain" Smart Dust (i.e., without using the Lighthouse Location System). Hence, the presented location system does not introduce additional restriction with respect to LOS.

Temporary LOS obstructions can cause wrong position estimates if a dust node misses one of the laser beams. However, the probability of such errors can be reduced by comparing two or more consecutive positions estimates and rejecting them if they differ by more than a small threshold. Reflected laser beams are typically not detected by the receiver hardware, since diffuse reflection reduces the laser light intensity drastically.

Note that other localization systems based on ultrasound and radio waves provide location estimates even in the case of an obstructed line-of-sight. However, the resulting location estimates are typically wrong due to relying on signals reflected around the obstruction. Often it is difficult to detect such situations [14], which may result in using wrong location estimates unnoticed.

4.5.6 Robustness

We assume that base stations are immobile and mounted in a safe place (with respect to harmful environmental influences) due to their potential long range (see Section 4.5.2). On the other hand, dust nodes are subject to mobility and other kinds of environmental influences (e.g., LOS obstructions), which can cause faulty location estimates.

However, in Sections 4.5.4 and 4.5.5 we mentioned extensions to the basic system in order to detect and reject such faulty locations estimates with high probability. This leaves us in a situation, where dust nodes either obtain good position estimates or no at all.

5 Related Work

Research has developed numerous systems and technologies for automatically locating people, equipment, and other tangibles. [16] gives an excellent overview and taxonomy of such location systems. These systems all involve gathering data by sensing real-world physical quantities. The data is in turn used to compute a location estimate. Common systems use diffuse infrared light [26, 27, 29], visible light [8, 9, 33], laser light [22, 23], ultrasound [12, 13, 15, 24, 25, 30], and radio waves [2, 3, 4, 18].

Some systems have been specifically designed for use in multi-hop wireless ad hoc and sensor networks [3, 4, 6, 10,

13, 25] and do not require any external hardware infrastructure besides the nodes of the network itself. Other systems rely on an external infrastructure typically consisting of many devices, which have to be carefully placed in the environment of the objects being located [2, 15, 24, 29, 30].

However, the special characteristics of future Smart Dust systems as described in Section 2 and the resulting requirements for a location system as described in Section 3 rule out the usage of all of these location systems. The small size and limited resources rule out systems based on radio waves and ultrasound, since transducers for these physical media are too large and transceivers consume too much energy for Smart Dust nodes [19]. The employed optical single-hop communication rules out all systems which require neighbor-to-neighbor or multi-hop communication. The use of a single (or few) base station(s) rules out all system which require substantial external infrastructure. The required localized location computation (see Section 3.2) rules out all systems where nodes cannot compute their location on their own. Additionally, many systems (e.g., ones based on vision and broadband ultrasound) typically have a high processing overhead and large memory footprint due to the necessary signal processing on the raw input data (e.g., time series of images or audio samples).

The systems that come closest to fulfilling the requirements of Smart Dust are ones based on vision or laser ranging techniques. Laser ranging systems are based on measuring the distance between the laser ranger device and some passive object by a variety of different methods [35]. However, with all these methods, only the active laser ranger can estimate the distance, not the object being located, which precludes localized location computation. The same is true for vision based methods, where a high resolution video camera is used to estimate node location [8]. There are also systems which combine laser ranging and vision-based methods [23], which obviously suffer from the same problem.

6 Conclusion

Future Smart Dust systems present a novel set of challenges to a location system. We examined these challenges and found that location systems developed in the past for mobile computing systems and COTS sensor networks are not applicable to Smart Dust systems due to the novel characteristics of the latter.

We presented the Lighthouse location system for future Smart Dust systems. By extending the base station, this system allows Smart Dust to autonomously estimate their physical location with respect to the base station with high precision over distances of tens of meters without node calibration. Besides the single modified base station, the system does not require any additional infrastructure components. This is achieved by

a new cylindrical lateration method. In contrast to traditional spherical methods, this approach does not have a wide baseline requirement. On the receiver side, only a simple optical receiver (amplified photo diode), moderate processing capabilities, and little memory are needed. That is, only marginal changes to the Smart Dust prototype developed at UC Berkeley are required.

We presented a prototype implementation of the system, a set of initial measurements, and a first analysis of several aspects of the system. Currently we are working on better support for node mobility and MMSE-based calibration. We are also currently analysing the system in more detail.

We plan to build a second revision of the base station prototype based on deflectable MEMS mirrors, which is expected to feature much improved accuracy, size, and power consumption.

Future work also includes an analysis of how a real Smart Dust implementation influences the quality of the location estimates. This includes factors like reduced clock resolution, increased clock skew, and the approximations mentioned in Section 4.5.3.

7 Acknowledgements

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, March 2002.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In *Infocom 2000*, Tel-Aviv, Israel, March 2000.

[3] J. Beutel. Geolocation in a PicoRadio Environment. Master's thesis, UC Berkeley, 1999.

[4] N. Bulusu, J. Heideman, and D. Estrin. GPS-less Low Cost Outdoor Localization for Very Small Devices. *IEEE Personal Communications*, 7(5):28–34, October 2000.

[5] N. Bulusu, J. Heideman, and D. Estrin. Adaptive Beacon Placement. In *ICDCS 2001*, Phoenix, USA, April 2001.

[6] S. Capkun, M. Hamdi, and J. P. Hubaux. GPS-free Positioning in Mobile Ad Hoc Networks. In *34th International Conference on System Sciences*, Hawaii, January 2001.

[7] R. Conant, J. Nee, K. Lau, and R. Muller. A Fast Flat Scanning Micromirror. In *2000 Solid-State Sensor and Actuator Workshop*, Hilton Head, USA, June 2000.

[8] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated Person Tracking Using Stereo, Color, and Pattern Recognition. In *Conference on Computer Vision and Pattern Recognition*, Santa Barbara, USA, June 1998.

[9] D. L. de Ipina. Video-based Sensing for Wide Deployment of Sentient Spaces. In *Workshop on Ubiquitous Computing and Communications*, Barcelona, Spain, September 2001.

[10] L. Doherty, K. S. J. Pister, and L. E. Ghaoui. Convex Position Estimation in Wireless Sensor Networks. In *Infocom 2001*, Anchorage, Alaska, April 2001.

[11] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the Physical World with Pervasive Networks. *IEEE Pervasive Computing*, 1(1):59–69, January 2002.

[12] E. Foxlin, M. Harrington, and G. Pfeifer. A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. In *25th Annual Conference on Computer Graphics*, Orlando, USA, July 1998.

[13] L. Girod, V. Bychkovskiy, J. Elson, and D. Estrin. Locating Tiny Sensors in Time and Space: A Case Study. In *International Conference on Computer Design (ICCD) 2002*, Freiburg, Germany, September 2002.

[14] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS) 2001*, Maui, Hawaii, October 2001.

[15] M. Hazas and A. Ward. A Novel Broadband Ultrasonic Location System. In *Ubicomp 2002*, Gothenburg, Sweden, September 2002.

[16] J. Hightower and G. Borriello. Location Systems for Ubiquitous Computing. *IEEE Computer*, 34(8):57–66, August 2001.

[17] J. Hightower, C. Vakili, G. Borriello, and R. Want. Design and Calibration of the SpotON Ad-Hoc Location Sensing System. unpublished, August 2001.

[18] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice, 4th Edition*. Springer-Verlag, 1997.

[19] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Emerging Challenges: Mobile Networking for Smart Dust. *Journal of Communications and Networks*, 2(3):188–196, September 2000.

[20] M. Langheinrich. Privacy by Design – Principles of Privacy-Aware Ubiquitous Systems. In *International Conference on Ubiquitous Computing (Ubicomp 01)*, Atlanta, GA, September 2001.

[21] M. Last and K. S. J. Pister. 2-DOF Actuated Micromirror Designed for Large DC Deflection. In *3rd Intl. Conf. on Micro Opto Electro Mechanical Systems (MEOMS 99)*, Mainz, Germany, August 1999.

[22] U. Lohr. Digital Elevation Models by Laserscanning: Principle and Applications. In *3rd Intl. Airborne Remote Sensing Conference*, Copenhagen, Denmark, 1997.

[23] B. Nickerson, P. Jasiobedzki, D. Wilkes, M. Jenkin, E. Milios, J. Tsotsos, A. Jepson, and O. N. Bains. The ARK Project: Autonomous Mobile Robots for Known Industrial Environments. *Robotics and Autonomous Systems*, 1(25):83–104, 1998.

[24] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *Mobicom 2000*, Boston, USA, August 2000.

[25] A. Savvides, C. C. Han, and M. Srivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *Mobicom 2001*, Rome, Italy, July 2001.

[26] B. N. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The ParcTab Mobile Computing System. In *4th Workshop on Workstation Operating Systems*, Napa, USA, October 1993.

[27] T. Starner, D. Kirsh, and S. Assefa. The Locust Swarm: An Environmentally Powered, Networkless Location and Messaging System. In *ISWC 1997*, Cambridge, USA, October 1997.

[28] H. Wang, D. Estrin, and L. Girod. Preprocessing in a Tiered Sensor Network for Habitat Monitoring. Submitted for publication, September 2002.

[29] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[30] A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[31] B. Warneke, M. Last, B. Leibowitz, and K. S. J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *IEEE Computer Magazine*, 34(1):44–51, January 2001.

[32] K. Whitehouse and D. Culler. Calibration as Parameter Estimation in Sensor Networks. In *Workshop on Wireless Sensor Networks and Applications (WSNA) 02*, Atlanta, USA, September 2002.

[33] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Realtime Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[34] FTB Feintechnik Bertsch. www.ftb-bertsch.de.

[35] Laser Distance Measurements. cord.org/cm/leot/Module6/module6.htm.

[36] National Semiconductors. www.national.com.

[37] parapin - a Parallel Port Programming Library for Linux. www.circlemud.org/~jelson/software/parapin/.

[38] Smart-Its Project. www.smart-its.org.