StarDust: A Flexible Architecture for Passive Localization in Wireless Sensor Networks [*]

Radu Stoleru, Pascal Vicaire, Tian He[†], John A. Stankovic

Department of Computer Science, University of Virginia
[†]Department of Computer Science and Engineering, University of Minnesota
{stoleru, pv9f}@cs.virginia.edu, tianhe@cs.umn.edu, stankovic@cs.virginia.edu

Abstract

The problem of localization in wireless sensor networks where nodes do not use ranging hardware, remains a challenging problem, when considering the required location accuracy, energy expenditure and the duration of the localization phase. In this paper we propose a framework, called StarDust, for wireless sensor network localization based on passive optical components. In the StarDust framework, sensor nodes are equipped with optical retro-reflectors. An aerial device projects light towards the deployed sensor network, and records an image of the reflected light. An image processing algorithm is developed for obtaining the locations of sensor nodes. For matching a node ID to a location we propose a constraint-based label relaxation algorithm. We propose and develop localization techniques based on four types of constraints: node color, neighbor information, deployment time for a node and deployment location for a node. We evaluate the performance of a localization system based on our framework by localizing a network of 26 sensor nodes deployed in a $120 \times 60\,ft^2$ area. The localization accuracy ranges from $2\,ft$ to $5\,ft$ while the localization time ranges from 10 milliseconds to 2 minutes.

Categories and Subject Descriptors: C.2.4 [Computer Communications Networks]: Distributed Systems; C.3 [Special Purpose and Application Based Systems]: Real-time and embedded systems

General Terms: Algorithms, Measurement, Performance, Design, Experimentation

Keywords: Wireless Sensor Networks, Localization

1 Introduction

Wireless Sensor Networks (WSN) have been envisioned to revolutionize the way humans perceive and interact with the surrounding environment. One vision is to embed tiny sensor devices in outdoor environments, by aerial deployments from unmanned air vehicles. The sensor nodes form a network and collaborate (to compensate for the extremely scarce resources available to each of them: computational power, memory size, communication capabilities) to accomplish the mission. Through collaboration, redundancy and fault tolerance, the WSN is then able to achieve unprecedented sensing capabilities.

A major step forward has been accomplished by developing systems for several domains: military surveillance [1] [2] [3], habitat monitoring [4] and structural monitoring [5]. Even after these successes, several research problems remain open. Among these open problems is sensor node localization, i.e., how to find the physical position of each sensor node. Despite the attention the localization problem in WSN has received, no universally acceptable solution has been developed. There are several reasons for this. On one hand, localization schemes that use ranging are typically high end solutions. GPS ranging hardware consumes energy, it is relatively expensive (if high accuracy is required) and poses form factor challenges that move us away from the vision of dust size sensor nodes. Ultrasound has a short range and is highly directional. Solutions that use the radio transceiver for ranging either have not produced encouraging results (if the received signal strength indicator is used) or are sensitive to environment (e.g., multipath). On the other hand, localization schemes that only use the connectivity information for inferring location information are characterized by low accuracies: $\approx 10\,ft$ in controlled environments, $40 - 50\,ft$ in realistic ones.

To address these challenges, we propose a framework for WSN localization, called StarDust, in which the complexity associated with the node localization is completely removed from the sensor node. The basic principle of the framework is localization through passivity: each sensor node is equipped with a corner-cube retro-reflector and possibly an optical filter (a coloring device). An aerial vehicle projects light onto the deployment area and records images containing retro-reflected light beams (they appear as luminous spots). Through image processing techniques, the locations of the retro-reflectors (i.e., sensor nodes) is deter-

mined. For inferring the identity of the sensor node present at a particular location, the StarDust framework develops a constraint-based node ID relaxation algorithm.

The main contributions of our work are the following. We propose a novel framework for node localization in WSNs that is very promising and allows for many future extensions and more accurate results. We propose a constraint-based label relaxation algorithm for mapping node IDs to the locations, and four constraints (node, connectivity, time and space), which are building blocks for very accurate and very fast localization systems. We develop a sensor node hardware prototype, called a SensorBall. We evaluate the performance of a localization system for which we obtain location accuracies of $2 - 5 ft$ with a localization duration ranging from 10 milliseconds to 2 minutes. We investigate the range of a system built on our framework by considering realities of physical phenomena that occurs during light propagation through the atmosphere.

The rest of the paper is structured as follows. Section 2 is an overview of the state of art. The design of the Star-Dust framework is presented in Section 3. One implementation and its performance evaluation are in Sections 4 and 5, followed by a suite of system optimization techniques, in Section 6. In Section 7 we present our conclusions.

2   Related Work

We present the prior work in localization in two major categories: the range-based, and the range-free schemes.

The range-based localization techniques have been designed to use either more expensive hardware (and hence higher accuracy) or just the radio transceiver. Ranging techniques dependent on hardware are the time-of-flight (ToF) and the time-difference-of-arrival (TDoA). Solutions that use the radio are based on the received signal strength indicator (RSSI) and more recently on radio interferometry.

The ToF localization technique that is most widely used is the GPS. GPS is a costly solution for a high accuracy localization of a large scale sensor network. AHLoS [6] employs a TDoA ranging technique that requires extensive hardware and solves relatively large nonlinear systems of equations. The Cricket location-support system (TDoA) [7] can achieve a location granularity of tens of inches with highly directional and short range ultrasound transceivers. In [2] the location of a sniper is determined in an urban terrain, by using the TDoA between an acoustic wave and a radio beacon. The PushPin project [8] uses the TDoA between ultrasound pulses and light flashes for node localization. The RADAR system [9] uses the RSSI to build a map of signal strengths as emitted by a set of beacon nodes. A mobile node is located by the best match, in the signal strength space, with a previously acquired signature. In MAL [10], a mobile node assists in measuring the distances (acting as constraints) between nodes until a rigid graph is generated. The localization problem is formulated as an on-line state estimation in a nonlinear dynamic system [11]. A cooperative ranging that attempts to achieve a global positioning from distributed local optimizations is proposed in [12]. A very recent, remarkable, localization technique is based on radio interferometry, RIPS [13], which utilizes two transmitters to create an interfering

signal. The frequencies of the emitters are very close to each other, thus the interfering signal will have a low frequency envelope that can be easily measured. The ranging technique performs very well. The long time required for localization and multi-path environments pose significant challenges.

Real environments create additional challenges for the range based localization schemes. These have been emphasized by several studies [14] [15] [16]. To address these challenges, and others (hardware cost, the energy expenditure, the form factor, the small range, localization time), several range-free localization schemes have been proposed. Sensor nodes use primarily connectivity information for inferring proximity to a set of anchors. In the Centroid localization scheme [17], a sensor node localizes to the centroid of its proximate beacon nodes. In APIT [18] each node decides its position based on the possibility of being inside or outside of a triangle formed by any three beacons within node's communication range. The Gradient algorithm [19], leverages the knowledge about the network density to infer the average one hop length. This, in turn, can be transformed into distances to nodes with known locations. DV-Hop [20] uses the hop by hop propagation capability of the network to forward distances to landmarks. More recently, several localization schemes that exploit the sensing capabilities of sensor nodes, have been proposed. Spotlight [21] creates well controlled (in time and space) events in the network while the sensor nodes detect and timestamp this events. From the spatio-temporal knowledge for the created events and the temporal information provided by sensor nodes, nodes' spatial information can be obtained. In a similar manner, the Lighthouse system [22] uses a parallel light beam, that is emitted by an anchor which rotates with a certain period. A sensor node detects the light beam for a period of time, which is dependent on the distance between it and the light emitting device.

Many of the above localization solutions target specific sets of requirements and are useful for specific applications. StarDust differs in that it addresses a particular demanding set of requirements that are not yet solved well. StarDust is meant for localizing air dropped nodes where node passiveness, high accuracy, low cost, small form factor and rapid localization are all required. Many military applications have such requirements.

3   StarDust System Design

The design of the StarDust system (and its name) was inspired by the similarity between a deployed sensor network, in which sensor nodes indicate their presence by emitting light, and the Universe consisting of luminous and illuminated objects: stars, galaxies, planets, etc.

The main difficulty when applying the above ideas to the real world is the complexity of the hardware that needs to be put on a sensor node so that the emitted light can be detected from thousands of feet. The energy expenditure for producing an intense enough light beam is also prohibitive.

Instead, what we propose to use for sensor node localization is a passive optical element called a retro-reflector. The most common retro-reflective optical component is a Corner-Cube Retroreflector (CCR), shown in Figure 1(a). It consists of three mutually perpendicular mirrors. The inter-
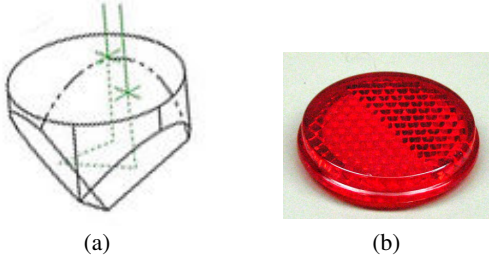
Figure 1. Corner-Cube Retroreflector (a) and an array of CCRs molded in plastic (b)

Figure 2. The StarDust system architecture

esting property of this optical component is that an incoming beam of light is reflected back, towards the source of the light, irrespective of the angle of incidence. This is in contrast with a mirror, which needs to be precisely positioned to be perpendicular to the incident light. A very common and inexpensive implementation of an array of CCRs is the retro-reflective plastic material used on cars and bicycles for night time detection, shown in Figure 1(b).
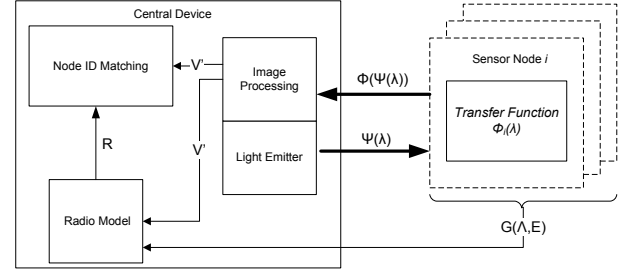
In the StarDust system, each node is equipped with a small (e.g. $0.5in^2$) array of CCRs and the enclosure has self-righting capabilities that orient the array of CCRs predominantly upwards. It is critical to understand that the upward orientation does not need to be exact. Even when large angular variations from a perfectly upward orientation are present, a CCR will return the light in the exact same direction from which it came.

In the remaining part of the section, we present the architecture of the StarDust system and the design of its main components.

3.1   System Architecture

The envisioned sensor network localization scenario is as follows:

- The sensor nodes are released, possibly in a controlled manner, from an aerial vehicle during the night.

- The aerial vehicle hovers over the deployment area and uses a strobe light to illuminate it. The sensor nodes, equipped with CCRs and optical filters (acting as coloring devices) have self-righting capabilities and retro-reflect the incoming strobe light. The retro-reflected light is either "white", as the originating source light, or colored, due to optical filters.

- The aerial vehicle records a sequence of two images very close in time (msec level). One image is taken when the strobe light is on, the other when the strobe light is off. The acquired images are used for obtaining the locations of sensor nodes (which appear as luminous spots in the image).

- The aerial vehicle executes the mapping of node IDs to the identified locations in one of the following ways: a) by using the color of a retro-reflected light, if a sensor node has a unique color; b) by requiring sensor nodes to establish neighborhood information and report it to a base station; c) by controlling the time sequence of sensor nodes deployment and recording additional im-

ages; d) by controlling the location where a sensor node is deployed.

- The computed locations are disseminated to the sensor network.

The architecture of the StarDust system is shown in Figure 2. The architecture consists of two main components: the first is centralized and it is located on a more powerful device. The second is distributed and it resides on all sensor nodes. The Central Device consists of the following: the Light Emitter, the Image Processing module, the Node ID Mapping module and the Radio Model. The distributed component of the architecture is the Transfer Function, which acts as a filter for the incoming light. The aforementioned modules are briefly described below:

- Light Emitter - It is a strobe light, capable of producing very intense, collimated light pulses. The emitted light is non-monochromatic (unlike a laser) and it is characterized by a spectral density $\Psi(\lambda)$, a function of the wavelength. The emitted light is incident on the CCRs present on sensor nodes.

- Transfer Function $\Phi(\Psi(\lambda))$ - This is a bandpass filter for the incident light on the CCR. The filter allows a portion of the original spectrum, to be retro-reflected. From here on, we will refer to the transfer function as the color of a sensor node.

- Image Processing - The Image Processing module acquires high resolution images. From these images the locations and the colors of sensor nodes are obtained. If only one set of pictures can be taken (i.e., one location of the light emitter/image analysis device), then the map of the field is assumed to be known as well as the distance between the imaging device and the field. The aforementioned assumptions (field map and distance to it) are not necessary if the images can be simultaneously taken from different locations. It is important to remark here that the identity of a node can not be directly obtained through Image Processing alone, unless a specific characteristic of a sensor node can be identified in the image.

- Node ID Matching - This module uses the detected locations and through additional techniques (e.g., sensor node coloring and connectivity information ($G(\Lambda, E)$) from the deployed network) to uniquely identify the sensor nodes observed in the image. The connectivity information is represented by neighbor tables sent from

Algorithm 1 Image Processing

1: Background filtering
2: Retro-reflected light recognition through intensity filtering
3: Edge detection to obtain the location of sensor nodes
4: Color identification for each detected sensor node

---

each sensor node to the Central Device.

- Radio Model - This component provides an estimate of the radio range to the Node ID Matching module. It is only used by node ID matching techniques that are based on the radio connectivity in the network. The estimate of the radio range R is based on the sensor node density (obtained through the Image Processing module) and the connectivity information (i.e., $G(\Lambda, E)$).

The two main components of the StarDust architecture are the Image Processing and the Node ID Mapping. Their design and analysis is presented in the sections that follow.

3.2 Image Processing

The goal of the Image Processing Algorithm (IPA) is to identify the location of the nodes and their color. Note that IPA does not identify which node fell where, but only what is the set of locations where the nodes fell.

IPA is executed after an aerial vehicle records two pictures: one in which the field of deployment is illuminated and one when no illuminations is present. Let P_{dark} be the picture of the deployment area, taken when no light was emitted and P_{light} be the picture of the same deployment area when a strong light beam was directed towards the sensor nodes.

The proposed IPA has several steps, as shown in Algorithm 1. The first step is to obtain a third picture P_{filter} where only the differences between P_{dark} and P_{light} remain. Let us assume that P_{dark} has a resolution of $n \times m$, where n is the number of pixels in a row of the picture, while m is the number of pixels in a column of the picture. Then P_{dark} is composed of $n \times m$ pixels noted $P_{dark}(i,j)$, $i \in 1 \le i \le n, 1 \le j \le m$. Similarly P_{light} is composed of $n \times m$ pixels noted $P_{light}(i,j)$, $1 \le i \le n, 1 \le j \le m$.

Each pixel P is described by an RGB value where the R value is denoted by P^R, the G value is denoted by P^G, and the B value is denoted by P^B. IPA then generates the third picture, P_{filter}, through the following transformations:

$$
\begin{aligned}
P^R_{filter}(i,j) &= P^R_{light}(i,j) - P^R_{dark}(i,j) \\
P^G_{filter}(i,j) &= P^G_{light}(i,j) - P^G_{dark}(i,j) \\
P^B_{filter}(i,j) &= P^B_{light}(i,j) - P^B_{dark}(i,j)
\end{aligned}
\tag{1}
$$

After this transformation, all the features that appeared in both P_{dark} and P_{light} are removed from P_{filter}. This simplifies the recognition of light retro-reflected by sensor nodes.

The second step consists of identifying the elements contained in P_{filter} that retro-reflect light. For this, an intensity filter is applied to P_{filter}. First IPA converts P_{filter} into a grayscale picture. Then the brightest pixels are identified and used to create $P_{reflect}$. This step is eased by the fact that the reflecting nodes should appear much brighter than any other illuminated object in the picture.
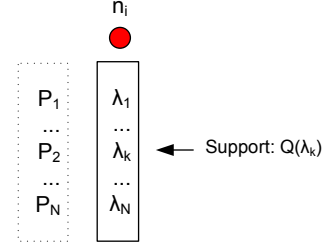
Figure 3. Probabilistic label relaxation

The third step runs an edge detection algorithm on $P_{reflect}$ to identify the boundary of the nodes present. A tool such as Matlab provides a number of edge detection techniques. We used the *bwboundaries* function. For the obtained edges, the location (x, y) (in the image) of each node is determined by computing the centroid of the points constituting its edges. Standard computer graphics techniques [23] are then used to transform the 2D locations of sensor nodes detected in multiple images into 3D sensor node locations. The color of the node is obtained as the color of the pixel located at (x, y) in P_{light}.

3.3 Node ID Matching

The goal of the Node ID Matching module is to obtain the identity (node ID) of a luminous spot in the image, detected to be a sensor node. For this, we define $V = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$ to be the set of locations of the sensor nodes, as detected by the Image Processing module and $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_m\}$ to be the set of unique node IDs assigned to the m sensor nodes, before deployment. From here on, we refer to node IDs as labels.

We model the problem of finding the label λ_j of a node n_i as a probabilistic label relaxation problem, frequently used in image processing/understanding. In the image processing domain, scene labeling (i.e., identifying objects in an image) plays a major role. The goal of scene labeling is to assign a label to each object detected in an image, such that an appropriate image interpretation is achieved. It is prohibitively expensive to consider the interactions among all the objects in an image. Instead, constraints placed among nearby objects generate local consistencies and through iteration, global consistencies can be obtained.

The main idea of the sensor node localization through probabilistic label relaxation is to iteratively compute the probability of each label being the correct label for a sensor node, by taking into account, at each iteration, the "support" for a label. The support for a label can be understood as a hint or proof, that a particular label is more likely to be the correct one, when compared with the other potential labels for a sensor node. We pictorially depict this main idea in Figure 3. As shown, node n_i has a set of candidate labels $\{\lambda_1, ..., \lambda_k\}$. Each of the labels has a different value for the Support function $Q(\lambda_k)$. We defer the explanation of how the Support function is implemented until the subsections that follow, where we provide four concrete techniques. Formally, the algorithm is outlined in Algorithm 2, where the equations necessary for computing the new probability $P_{n_i}(\lambda_k)$ for a label λ_k of a node n_i, are expressed by the

Algorithm 2 Label Relaxation

1: **for** each sensor node n_i **do**
2:     assign equal prob. to all possible labels
3: **end for**
4: **repeat**
5:     $converged \leftarrow true$
6:     **for** each sensor node n_i **do**
7:         **for** each each label λ_j of n_i **do**
8:             compute the Support label λ_j: Equation 4
9:         **end for**
10:         compute K for the node n_i: Equation 3
11:         **for** each each label λ_j **do**
12:             update probability of label λ_j: Equation 2
13:             **if** $|new\ prob. - old\ prob.| \geq \varepsilon$ **then**
14:                 $converged \leftarrow false$
15:             **end if**
16:         **end for**
17:     **end for**
18: **until** $converged = true$

following equations:

$$P_{n_i}^{s+1}(\lambda_k) = \frac{1}{K_{n_i}} P_{n_i}^s(\lambda_k) Q_{n_i}^s(\lambda_k) \qquad (2)$$

where K_{n_i} is a normalizing constant, given by:

$$K_{n_i} = \sum_{k=1}^{N} P_{n_i}^s(\lambda_k) Q_{n_i}^s(\lambda_k) \qquad (3)$$

and $Q_{n_i}^s(\lambda_k)$ is:

$$Q_{n_i}^s(\lambda_k) = \text{``support for label } \lambda_k \text{ of node } n_i\text{''} \qquad (4)$$

The label relaxation algorithm is iterative and it is polynomial in the size of the network(number of nodes). The pseudo-code is shown in Algorithm 2. It initializes the probabilities associated with each possible label, for a node n_i, through a uniform distribution. At each iteration s, the algorithm updates the probability associated with each label, by considering the Support $Q_{n_i}^s(\lambda_k)$ for each candidate label of a sensor node.

In the sections that follow, we describe four different techniques for implementing the Support function: based on node coloring, radio connectivity, the time of deployment (time) and the location of deployment (space). While some of these techniques are simplistic, they are primitives which, when combined, can create powerful localization systems. These design techniques have different trade-offs, which we will present in Section 3.3.6.

3.3.1 Relaxation with Color Constraints

The unique mapping between a sensor node's position (identified by the image processing) and a label can be obtained by assigning a unique color to each sensor node. For this we define $C = \{c_1, c_2, ..., c_n\}$ to be the set of unique colors available and $M : \Lambda \to C$ to be a one-to-one mapping of labels to colors. This mapping is known prior to the sensor node deployment (from node manufacturing).

In the case of color constrained label relaxation, the support for label λ_k is expressed as follows:

$$Q_{n_i}^s(\lambda_k) = 1 \qquad (5)$$

As a result, the label relaxation algorithm (Algorithm 2) consists of the following steps: one label is assigned to each sensor node (lines 1-3 of the algorithm), implicitly having a probability $P_{n_i}(\lambda_k) = 1$; the algorithm executes a single iteration, when the support function, simply, reiterates the confidence in the unique labeling.

However, it is often the case that unique colors for each node will not be available. It is interesting to discuss here the influence that the size of the coloring space (i.e., $|C|$) has on the accuracy of the localization algorithm. Several cases are discussed below:

- If $|C| = 0$, no colors are used and the sensor nodes are equipped with simple CCRs that reflect back all the incoming light (i.e., no filtering, and no coloring of the incoming light). From the image processing system, the position of sensor nodes can still be obtained. Since all nodes appear white, no single sensor node can be uniquely identified.

- If $|C| = m - 1$ then there are enough unique colors for all nodes (one node remains white, i.e. no coloring), the problem is trivially solved. Each node can be identified, based on its unique color. This is the scenario for the relaxation with color constraints.

- If $|C| \geq 1$, there are several options for how to partition the coloring space. If $C = \{c_1\}$ one possibility is to assign the color c_1 to a single node, and leave the remaining $m - 1$ sensor nodes white, or to assign the color c_1 to more than one sensor node. One can observe that once a color is assigned uniquely to a sensor node, in effect, that sensor node is given the status of "anchor", or node with known location.

It is interesting to observe that there is an entire spectrum of possibilities for how to partition the set of sensor nodes in equivalence classes (where an equivalence class is represented by one color), in order to maximize the success of the localization algorithm. One of the goals of this paper is to understand how the size of the coloring space and its partitioning affect localization accuracy.

Despite the simplicity of this method of constraining the set of labels that can be assigned to a node, we will show that this technique is very powerful, when combined with other relaxation techniques.

3.3.2 Relaxation with Connectivity Constraints

Connectivity information, obtained from the sensor network through beaconing, can provide additional information for locating sensor nodes. In order to gather connectivity information, the following need to occur: 1) after deployment, through beaconing of HELLO messages, sensor nodes build their neighborhood tables; 2) each node sends its neighbor table information to the Central device via a base station.

First, let us define $G = (\Lambda, E)$ to be the weighted connectivity graph built by the Central device from the received neighbor table information. In G the edge (λ_i, λ_j) has a
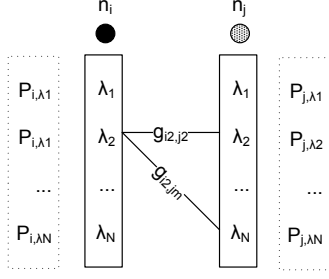
Figure 4. Label relaxation with connectivity constraints

weight g_{ij} represented by the number of beacons sent by λ_j and received by λ_i. In addition, let R be the radio range of the sensor nodes.

The main idea of the connectivity constrained label relaxation is depicted in Figure 4 in which two nodes n_i and n_j have been assigned all possible labels. The confidence in each of the candidate labels for a sensor node, is represented by a probability, shown in a dotted rectangle.

It is important to remark that through beaconing and the reporting of neighbor tables to the Central Device, a global view of all constraints in the network can be obtained. It is critical to observe that these constraints are among labels. As shown in Figure 4 two constraints exist between nodes n_i and n_j. The constraints are depicted by $g_{i2,j2}$ and $g_{i2,jM}$, the number of beacons sent the labels λ_{j2} and λ_{jM} and received by the label λ_{i2}.

The support for the label λ_k of sensor node n_i, resulting from the "interaction" (i.e., within radio range) with sensor node n_j is given by:

$$Q_{n_i}^s(\lambda_k) = \sum_{m=1}^{M} g_{\lambda_k \lambda_m} P_{n_j}^s(\lambda_m) \qquad (6)$$

As a result, the localization algorithm (Algorithm 3 consists of the following steps: all labels are assigned to each sensor node (lines 1-3 of the algorithm), and implicitly each label has a probability initialized to $P_{n_i}(\lambda_k) = 1/|\Lambda|$; in each iteration, the probabilities for the labels of a sensor node are updated, when considering the interaction with the labels of sensor nodes within R. It is important to remark that the identity of the nodes within R is not known, only the candidate labels and their probabilities. The relaxation algorithm converges when, during an iteration, the probability of no label is updated by more than ε.

The label relaxation algorithm based on connectivity constraints, enforces such constraints between pairs of sensor nodes. For a large scale sensor network deployment, it is not feasible to consider all pairs of sensor nodes in the network. Hence, the algorithm should only consider pairs of sensor nodes that are within a reasonable communication range (R). We assume a circular radio range and a symmetric connectivity. In the remaining part of the section we propose a simple analytical model that estimates the radio range R for medium-connected networks (less than 20 neighbors per R). We consider the following to be known: the size of the deployment field (L), the number of sensor nodes deployed (N)

Algorithm 3 Localization
1: Estimate the radio range R
2: Execute the Label Relaxation Algorithm with Support Function given by Equation 6 for neighbors less than R apart
3: **for** each sensor node n_i **do**
4:     node identity is λ_k with max. prob.
5: **end for**

---

and the total number of unidirectional (i.e., not symmetric) one-hop radio connections in the network (k). For our analysis, we uniformly distribute the sensor nodes in a square area of length L, by using a grid of unit length $L/\sqrt{N}$. We use the substitution $u = L/\sqrt{N}$ to simplify the notation, in order to distinguish the following cases: if $u \leq R \leq \sqrt{2}u$ each node has four neighbors (the expected $k = 4N$); if $\sqrt{2}u \leq R \leq 2u$ each node has eight neighbors (the expected $k = 8N$); if $2u \leq R \leq \sqrt{5}u$ each node has twelve neighbors (the expected $k = 12N$); if $\sqrt{5}u \leq R \leq 3u$ each node has twenty neighbors (the expected $k = 20N$)

For a given $t = k/4N$ we take R to be the middle of the interval. As an example, if $t = 5$ then $R = (3 + \sqrt{5})u/2$. A quadratic fitting for R over the possible values of t, produces the following closed-form solution for the communication range R, as a function of network connectivity k, assuming L and N constant:

$$R(k) = \frac{L}{\sqrt{N}} \left[-0.051 \left(\frac{k}{4N} \right)^2 + 0.66 \left(\frac{k}{4N} \right) + 0.6 \right] \quad (7)$$

We investigate the accuracy of our model in Section 5.2.1.

3.3.3 Relaxation with Time Constraints

Time constraints can be treated similarly with color constraints. The unique identification of a sensor node can be obtained by deploying sensor nodes individually, one by one, and recording a sequence of images. The sensor node that is identified as new in the last picture (it was not identified in the picture before last) must be the last sensor node dropped.

In a similar manner with color constrained label relaxation, the time constrained approach is very simple, but may take too long, especially for large scale systems. While it can be used in practice, it is unlikely that only a time constrained label relaxation is used. As we will see, by combining constrained-based primitives, realistic localization systems can be implemented.

The support function for the label relaxation with time constraints is defined identically with the color constrained relaxation:
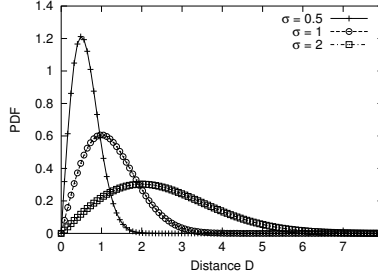
$$Q_{n_i}^s(\lambda_k) = 1 \qquad (8)$$

The localization algorithm (Algorithm 2 consists of the following steps: one label is assigned to each sensor node (lines 1-3 of the algorithm), and implicitly having a probability $P_{n_i}(\lambda_k) = 1$; the algorithm executes a single iteration,
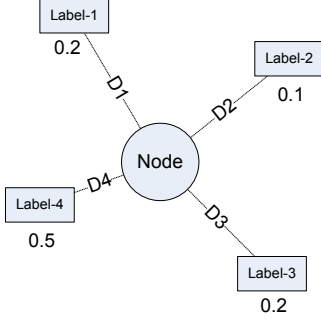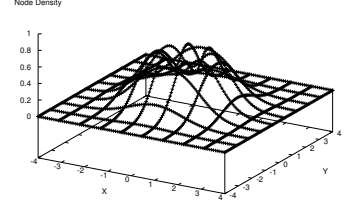
Figure 5. Relaxation with space constraints

Figure 6. Probability distribution of distances

Figure 7. Distribution of nodes

when the support function, simply, reiterates the confidence in the unique labeling.

3.3.4 Relaxation with Space Constraints

Spatial information related to sensor deployment can also be employed as another input to the label relaxation algorithm. To do that, we use two types of locations: the node location p_n and the label location p_l. The former p_n is defined as the position of nodes (x_n, y_n, z_n) after deployment, which can be obtained through Image Processing as mentioned in Section 3.3. The latter p_l is defined as the location (x_l, y_l, z_l) where a node is dropped. We use $D_{\lambda_m}^{n_i}$ to denote the horizontal distance between the location of the label λ_m and the location of the node n_i. Clearly, $D_{\lambda_m}^{n_i} = \sqrt{(x_n - x_l)^2 + (y_n - y_l)^2}$.

At the time of a sensor node release, the one-to-one mapping between the node and its label is known. In other words, the label location is the same as the node location at the release time. After release, the label location information is partially lost due to the random factors such as wind and surface impact. However, statistically, the node locations are correlated with label locations. Such correlation depends on the airdrop methods employed and environments. For the sake of simplicity, let's assume nodes are dropped from the air through a helicopter hovering in the air. Wind can be decomposed into three components $\vec{X}, \vec{Y}$ and $\vec{Z}$. Only $\vec{X}$ and $\vec{Y}$ affect the horizontal distance a node can travel. According to [24], we can assume that $\vec{X}$ and $\vec{Y}$ follow an independent normal distribution. Therefore, the absolute value of the wind speed follows a Rayleigh distribution. Obviously the higher the wind speed is, the further a node would land away horizontally from the label location. If we assume that the distance D is a function of the wind speed V [25] [26], we can obtain the probability distribution of D under a given wind speed distribution. Without loss of generality, we assume that D is proportional to the wind speed. Therefore, D follows the Rayleigh distribution as well. As shown in Figure 5, the spatial-based relaxation is a recursive process to assign the probability that a nodes has a certain label by using the distances between the location of a node with multiple label locations.

We note that the distribution of distance D affects the probability with which a label is assigned. It is not necessarily true that the nearest label is always chosen. For example, if D follows the Rayleigh(σ^2) distribution, we can obtain the

Probability Density Function (PDF) of distances as shown in Figure 6. This figure indicates that the possibility of a node to fall vertically is very small under windy conditions ($\sigma > 0$), and that the distance D is affected by the σ. The spatial distribution of nodes for $\sigma = 1$ is shown in Figure 7. Strong wind with a high σ value leads to a larger node dispersion. More formally, given a probability density function $PDF(D)$, the support for label λ_k of sensor node n_i can be formulated as:

$$Q_{n_i}^s(\lambda_k) = PDF(D_{\lambda_k}^{n_i}) \tag{9}$$

It is interesting to point out two special cases. First, if all nodes are released at once (i.e., only one label location for all released nodes), the distance D from a node to all labels is the same. In this case, $P_{n_i}^{s+1}(\lambda_k) = P_{n_i}^s(\lambda_k)$, which indicates that we can not use the spatial-based relaxation to recursively narrow down the potential labels for a node. Second, if nodes are released at different locations that are far away from each other, we have: (i) If node n_i has label λ_k, $P_{n_i}^s(\lambda_k) \rightarrow 1$ when $s \rightarrow \infty$, (ii) If node n_i does not have label λ_k, $P_{n_i}^s(\lambda_k) \rightarrow 0$ when $s \rightarrow \infty$. In this second scenario, there are multiple labels (one label per release), hence it is possible to correlate release times (labels) with positions on the ground. These results indicate that spatial-based relaxation can label the node with a very high probability if the physical separation among nodes is large.

3.3.5 Relaxation with Color and Connectivity Constraints

One of the most interesting features of the StarDust architecture is that it allows for hybrid localization solutions to be built, depending on the system requirements. One example is a localization system that uses the color and connectivity constraints. In this scheme, the color constraints are used for reducing the number of candidate labels for sensor nodes, to a more manageable value. As a reminder, in the connectivity constrained relaxation, all labels are candidate labels for each sensor node. The color constraints are used in the initialization phase of Algorithm 3 (lines 1-3). After the initialization, the standard connectivity constrained relaxation algorithm is used.

For a better understanding of how the label relaxation algorithm works, we give a concrete example, exemplified in Figure 8. In part (a) of the figure we depict the data structures

(a)   (b)
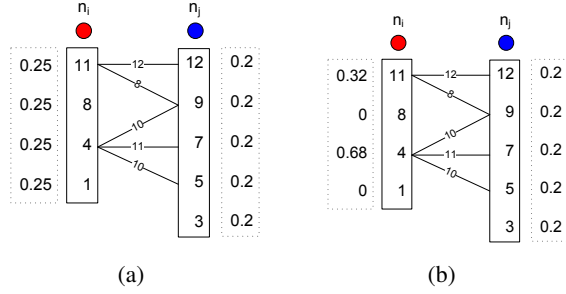
Figure 8. A step through the algorithm. After initialization (a) and after the 1st iteration for node n_i (b)

(a)   (b)

Figure 9. SensorBall with self-righting capabilities (a) and colored CCRs (b)

associated with nodes n_i and n_j after the initialization steps of the algorithm (lines 1-6), as well as the number of beacons between different labels (as reported by the network, through $G(\Lambda, E)$). As seen, the potential labels (shown inside the vertical rectangles) are assigned to each node. Node n_i can be any of the following: $11, 8, 4, 1$. Also depicted in the figure are the probabilities associated with each of the labels. After initialization, all probabilities are equal.

Part (b) of Figure 8 shows the result of the first iteration of the localization algorithm for node n_i, assuming that node n_j is the first w_i chosen in line 7 of Algorithm 3. By using Equation 6, the algorithm computes the "support" $Q(\lambda_i)$ for each of the possible labels for node n_i. Once the $Q(\lambda_i)$'s are computed, the normalizing constant, given by Equation 3 can be obtained. The last step of the iteration is to update the probabilities associated with all potential labels of node n_i, as given by Equation 2.

One interesting problem, which we explore in the performance evaluation section, is to assess the impact the partitioning of the color set C has on the accuracy of localization. When the size of the coloring set is smaller than the number of sensor nodes (as it is the case for our hybrid connectivity/color constrained relaxation), the system designer has the option of allowing one node to uniquely have a color (acting as an anchor), or multiple nodes. Intuitively, by assigning one color to more than one node, more constraints (distributed) can be enforced.

3.3.6 Relaxation Techniques Analysis

The proposed label relaxation techniques have different trade-offs. For our analysis of the trade-offs, we consider the following metrics of interest: the localization time (duration), the energy consumed (overhead), the network size (scale) that can be handled by the technique and the localization accuracy. The parameters of interest are the following: the number of sensor nodes (N), the energy spent for one aerial drop (ε_d), the energy spent in the network for collecting and reporting neighbor information ε_b and the time T_d taken by a sensor node to reach the ground after being aerially deployed. The cost comparison of the different label relaxation techniques is shown in Table 1.

As shown, the relaxation techniques based on color and space constraints have the lowest localization duration, zero, for all practical purposes. The scalability of the color based relaxation technique is, however, limited to the number of
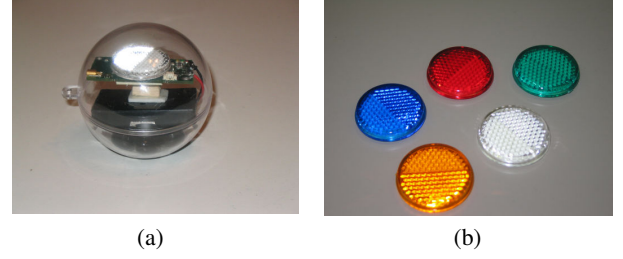
unique color filters that can be built. The narrower the Transfer Function $\Psi(\lambda)$, the larger the number of unique colors that can be created. The manufacturing costs, however, are increasing as well. The scalability issue is addressed by all other label relaxation techniques. Most notably, the time constrained relaxation, which is very similar to the color-constrained relaxation, addresses the scale issue, at a higher deployment cost.

Criteria	Color	Connectivity	Time	Space
Duration	0	NT_b	NT_d	0
Overhead	ε_d	$\varepsilon_d + N\varepsilon_b$	$N\varepsilon_d$	ε_d
Scale	$|C|$	$|N|$	$|N|$	$|N|$
Accuracy	High	Low	High	Medium

Table 1. Comparison of label relaxation techniques

4 System Implementation

The StarDust localization framework, depicted in Figure 2, is flexible in that it enables the development of new localization systems, based on the four proposed label relaxation schemes, or the inclusion of other, yet to be invented, schemes. For our performance evaluation we implemented a version of the StarDust framework, namely the one proposed in Section 3.3.5, where the constraints are based on color and connectivity.

The Central device of the StarDust system consists of the following: the Light Emitter - we used a common-off-the-shelf flash light (QBeam, 3 million candlepower); the image acquisition was done with a 3 megapixel digital camera (Sony DSC-S50) which provided the input to the Image Processing algorithm, implemented in Matlab.

For sensor nodes we built a custom sensor node, called SensorBall, with self-righting capabilities, shown in Figure 9(a). The self-righting capabilities are necessary in order to orient the CCR predominantly upwards. The CCRs that we used were inexpensive, plastic molded, night time warning signs commonly available on bicycles, as shown in Figure 9(b). We remark here the low quality of the CCRs we used. The reflectivity of each CCR (there are tens molded in the plastic container) is extremely low, and each CCR is not built with mirrors. A reflective effect is achieved by employing finely polished plastic surfaces. We had 5 colors available, in addition to the standard CCR, which reflects all the incoming light (white CCR). For a slightly higher price (ours were 20cents/piece), better quality CCRs can be employed.

Figure 10. The field in the dark

Figure 11. The illuminated field

Figure 12. The difference: Figure 10 - Figure 11

Higher quality (better mirrors) would translate in more accurate image processing (better sensor node detection) and smaller form factor for the optical component (an array of CCRs with a smaller area can be used).

The sensor node platform we used was the micaZ mote. The code that runs on each node is a simple application which broadcasts 100 beacons, and maintains a neighbor table containing the percentage of successfully received beacons, for each neighbor. On demand, the neighbor table is reported to a base station, where the node ID mapping is performed.

5   System Evaluation

In this section we present the performance evaluation of a system implementation of the StarDust localization framework. The three major research questions that our evaluation tries to answer are: the feasibility of the proposed framework (can sensor nodes be optically detected at large distances), the localization accuracy of one actual implementation of the StarDust framework, and whether or not atmospheric conditions can affect the recognition of sensor nodes in an image. The first two questions are investigated by evaluating the two main components of the StarDust framework: the Image Processing and the Node ID Matching. These components have been evaluated separately mainly because of lack of adequate facilities. We wanted to evaluate the performance of the Image Processing Algorithm in a long range, realistic, experimental set-up, while the Node ID Matching required a relatively large area, available for long periods of time (for connectivity data gathering). The third research question is investigated through a computer modeling of atmospheric phenomena.

For the evaluation of the Image Processing module, we performed experiments in a football stadium where we deploy 6 sensor nodes in a 3×2 grid. The distance between the Central device and the sensor nodes is approximately $500 ft$. The metrics of interest are the number of false positives and false negatives in the Image Processing Algorithm.

For the evaluation of the Node ID Mapping component, we deploy 26 sensor nodes in an $120 \times 60 ft^2$ flat area of a stadium. In order to investigate the influence the radio connectivity has on localization accuracy, we vary the height above ground of the deployed sensor nodes. Two set-ups are used: one in which the sensor nodes are on the ground, and

the second one, in which the sensor nodes are raised 3 inches above ground. From here on, we will refer to these two experimental set-ups as the low connectivity and the high connectivity networks, respectively because when nodes are on the ground the communication range is low resulting in less neighbors than when the nodes are elevated and have a greater communication range. The metrics of interest are: the localization error (defined as the distance between the computed location and the true location - known from the manual placement), the percentage of nodes correctly localized, the convergence of the label relaxation algorithm, the time to localize and the robustness of the node ID mapping to errors in the Image Processing module.

The parameters that we vary experimentally are: the angle under which images are taken, the focus of the camera, and the degree of connectivity. The parameters that we vary in simulations (subsequent to image acquisition and connectivity collection) the number of colors, the number of anchors, the number of false positives or negatives as input to the Node ID Matching component, the distance between the imaging device and sensor network (i.e., range), atmospheric conditions (light attenuation coefficient) and CCR reflectance (indicative of its quality).

5.1   Image Processing

For the IPA evaluation, we deploy 6 sensor nodes in a 3×2 grid. We take 13 sets of pictures using different orientations of the camera and different zooming factors. All pictures were taken from the same location. Each set is composed of a picture taken in the dark and of a picture taken with a light beam pointed at the nodes. We process the pictures offline using a Matlab implementation of IPA. Since we are interested in the feasibility of identifying colored sensor nodes at large distance, the end result of our IPA is the 2D location of sensor nodes (position in the image). The transformation to 3D coordinates can be done through standard computer graphics techniques [23].

One set of pictures obtained as part of our experiment is shown in Figures 10 and 11. The execution of our IPA algorithm results in Figure 12 which filters out the background, and Figure 13 which shows the output of the edge detection step of IPA. The experimental results are depicted in Figure 14. For each set of pictures the graph shows the number of false positives (the IPA determines that there is a node
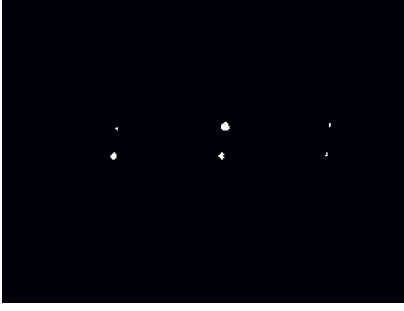
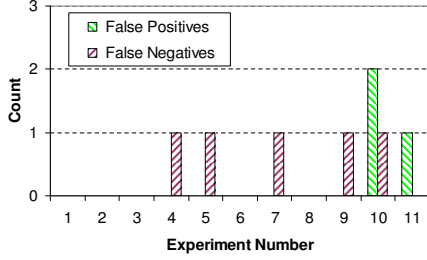**Figure 13. Retroreflectors detected in Figure 12**

Figure 14. False Positives and Negatives for the 6 nodes

Figure 15. The number of existing and missing radio connections in the sparse connectivity experiment

Figure 16. The number of existing and missing radio connections in the high connectivity experiment

while there is none), and the number of false negatives (the IPA determines that there is no node while there is one). In about 45% of the cases, we obtained perfect results, i.e., no false positives and no false negatives. In the remaining cases, we obtained a number of false positives of at most one, and a number of false negatives of at most two.

We exclude two pairs of pictures from Figure 14. In the first excluded pair, we obtain 42 false positives and in the second pair 10 false positives and 7 false negatives. By carefully examining the pictures, we realized that the first pair was taken out of focus and that a car temporarily appeared in one of the pictures of the second pair. The anomaly in the second set was due to the fact that we waited too long to take the second picture. If the pictures had been taken a few milliseconds apart, the car would have been represented on either both or none of the pictures and the IPA would have filtered it out.

5.2 Node ID Matching

We evaluate the Node ID Matching component of our system by collecting empirical data (connectivity information) from the outdoor deployment of 26 nodes in the $120 \times 60 ft^2$ area. We collect 20 sets of data for the high connectivity and low connectivity network deployments. Off-line we investigate the influence of coloring on the metrics of interest, by randomly assigning colors to the sensor nodes. For one experimental data set we generate 50 random assignments of colors to sensor nodes. It is important to observe that, for the evaluation of the Node ID Matching algorithm (color and connectivity constrained), we simulate the color assignment to sensor nodes. As mentioned in Section 4 the size of the coloring space available to us was 5 (5 colors). Through simulations of color assignment (not connectivity) we are able to investigate the influence that the size of the coloring space has on the accuracy of localization. The value of the param-
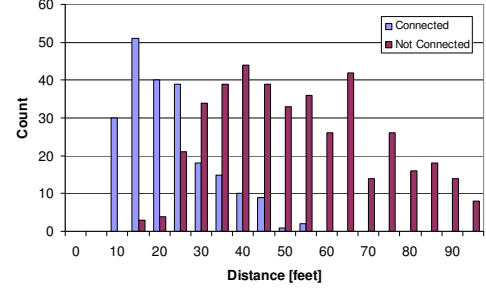
eter ε used in Algorithm 2 was 0.001. The results presented here represent averages over the randomly generated colorings and over all experimental data sets.
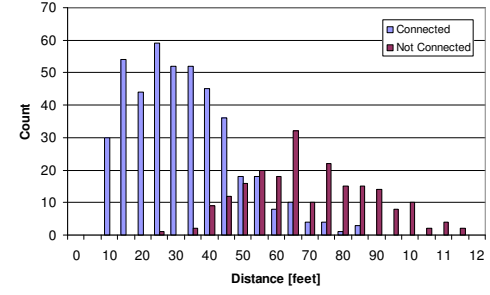
We first investigate the accuracy of our proposed Radio Model, and subsequently use the derived values for the radio range in the evaluation of the Node ID matching component.

5.2.1 Radio Model

From experiments, we obtain the average number of observed beacons (k, defined in Section 3.3.2) for the low connectivity network of 180 beacons and for the high connectivity network of 420 beacons. From our Radio Model (Equation 7, we obtain a radio range $R = 25 ft$ for the low connectivity network and $R = 40 ft$ for the high connectivity network.

To estimate the accuracy of our simple model, we plot the number of radio links that exist in the networks, and the number of links that are missing, as functions of the distance between nodes. The results are shown in Figures 15 and 16. We define the average radio range R to be the distance over which less than 20% of potential radio links, are missing. As shown in Figure 15, the radio range is between $20 ft$ and $25 ft$. For the higher connectivity network, the radio range was between $30 ft$ and $40 ft$.

We choose two conservative estimates of the radio range: $20 ft$ for the low connectivity case and $35 ft$ for the high connectivity case, which are in good agreement with the values predicted by our Radio Model.

5.2.2 Localization Error vs. Coloring Space Size

In this experiment we investigate the effect of the number of colors on the localization accuracy. For this, we randomly assign colors from a pool of a given size, to the sensor nodes.
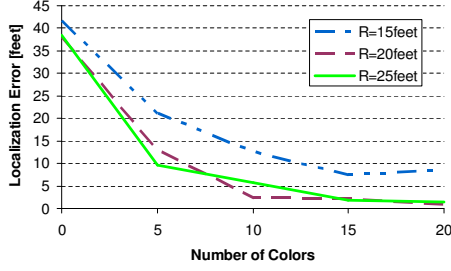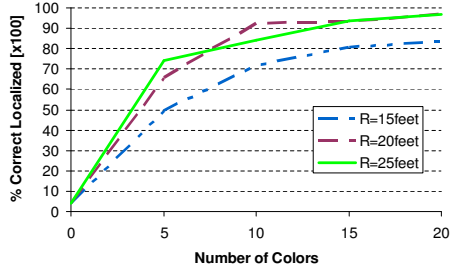
Figure 17. Localization error

Figure 19. Convergence error

Figure 18. Percentage of nodes correctly localized

Figure 20. Localization error vs. number of colors

We then execute the localization algorithm, which uses the empirical data. The algorithm is run for three different radio ranges: 15, 20 and $25 ft$, to investigate its influence on the localization error.

The results are depicted in Figure 17 (localization error) and Figure 18 (percentage of nodes correctly localized). As shown, for an estimate of $20 ft$ for the radio range (as predicted by our Radio Model) we obtain the smallest localization errors, as small as $2 ft$, when enough colors are used. Both Figures 17 and 18 confirm our intuition that a larger number of colors available significantly decrease the error in localization.

The well known fact that relaxation algorithms do not always converge, was observed during our experiments. The percentage of successful runs (when the algorithm converged) is depicted in Figure 19. As shown, in several situations, the algorithm failed to converge (the algorithm execution was stopped after 100 iterations per node). If the algorithm does not converge in a predetermined number of steps, it will terminate and the label with the highest probability will provide the identity of the node. It is very probable that the chosen label is incorrect, since the probabilities of some of labels are constantly changing (with each iteration).The convergence of relaxation based algorithms is a well known issue.

5.2.3 Localization Error vs. Color Uniqueness

As mentioned in the Section 3.3.1, a unique color gives a sensor node the statute of an anchor. A sensor node that is an anchor can unequivocally be identified through the Image Processing module. In this section we investigate the effect unique colors have on the localization accuracy. Specifically, we want to experimentally verify our intuition that assigning more nodes to a color can benefit the localization accuracy, by enforcing more constraints, as opposed to uniquely assigning a color to a single node.
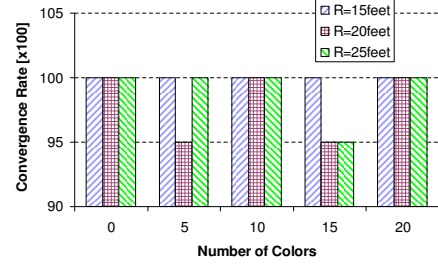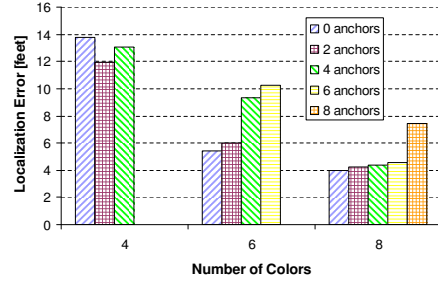
For this, we fix the number of available colors to either 4, 6 or 8 and vary the number of nodes that are given unique colors, from 0, up to the maximum number of colors (4, 6 or 8). Naturally, if we have a maximum number of colors of 4, we can assign at most 4 anchors. The experimental results are depicted in Figure 20 (localization error) and Figure 21 (percentage of sensor node correctly localized). As expected, the localization accuracy increases with the increase in the number of colors available (larger coloring space). Also, for a given size of the coloring space (e.g., 6 colors available), if more colors are uniquely assigned to sensor nodes then the localization accuracy decreases. It is interesting to observe that by assigning colors uniquely to nodes, the benefit of having additional colors is diminished. Specifically, if 8 colors are available and all are assigned uniquely, the system would be less accurately localized (error $\approx 7 ft$), when compared to the case of 6 colors and no unique assignments of colors ($\approx 5 ft$ localization error).

The same trend, of a less accurate localization can be observed in Figure 21, which shows the percentage of nodes correctly localized (i.e., $0 ft$ localization error). As shown, if we increase the number of colors that are uniquely assigned, the percentage of nodes correctly localized decreases.

5.2.4 Localization Error vs. Connectivity

We collected empirical data for two network deployments with different degrees of connectivity (high and low) in order to assess the influence of connectivity on location accuracy. The results obtained from running our localization algorithm are depicted in Figure 22 and Figure 23. We varied the number of colors available and assigned no anchors (i.e., no unique assignments of colors).

In both scenarios, as expected, localization error decrease with an increase in the number of colors. It is interesting to observe, however, that the low connectivity scenario im-
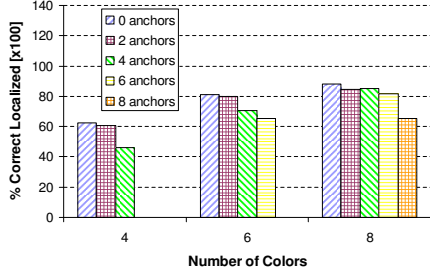
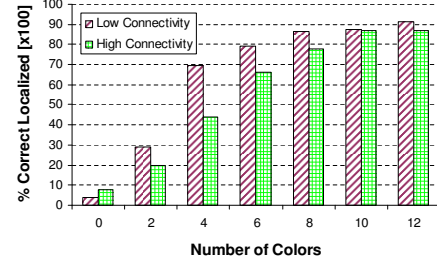**Figure 21. Percentage of nodes correctly localized vs. number of colors**
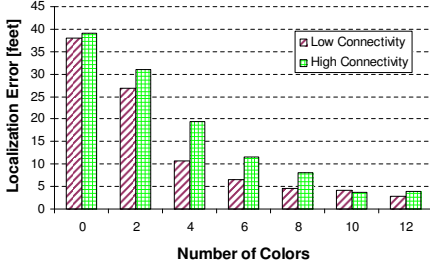
Figure 22. Localization error vs. number of colors

Figure 23. Percentage of nodes correctly localized

Figure 24. Impact of false negatives on the localization error

proves the localization accuracy quicker, from the additional number of colors available. When the number of colors becomes relatively large (twelve for our 26 sensor node network), both scenarios (low and high connectivity) have comparable localization errors, of less that $2ft$. The same trend of more accurate location information is evidenced by Figure 23 which shows that the percentage of nodes that are localized correctly grows quicker for the low connectivity deployment.
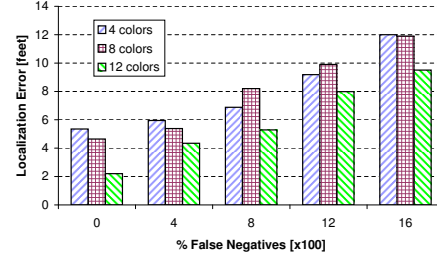
5.3 Localization Error vs. Image Processing Errors

So far we investigated the sources for error in localization that are intrinsic to the Node ID Matching component. As previously presented, luminous objects can be mistakenly detected to be sensor nodes during the location detection phase of the Image Processing module. These false positives can be eliminated by the color recognition procedure of the Image Processing module. More problematic are false negatives (when a sensor node does not reflect back enough light to be detected). They need to be handled by the localization algorithm. In this case, the localization algorithm is presented with two sets of nodes of different sizes, that need to be matched: one coming from the Image Processing (which misses some nodes) and one coming from the network, with the connectivity information (here we assume a fully connected network, so that all sensor nodes report their connectivity information). In this experiment we investigate how Image Processing errors (false negatives) influence the localization accuracy.

For this evaluation, we ran our localization algorithm with empirical data, but dropped a percentage of nodes from the list of nodes detected by the Image Processing algorithm (we artificially introduced false negatives in the Image Process-

ing). The effect of false negatives on localization accuracy is depicted in Figure 24. As seen in the figure if the number of false negatives is 15%, the error in position estimation doubles when 4 colors are available. It is interesting to observe that the scenario when more colors are available (e.g., 12 colors) is being affected more drastically than the scenario with less colors (e.g., 4 colors). The benefit of having more colors available is still being maintained, at least for the range of colors we investigated (4 through 12 colors).
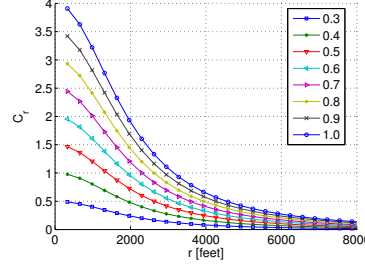
5.4 Localization Time

In this section we look more closely at the duration for each of the four proposed relaxation techniques and two combinations of them: color-connectivity and color-time. We assume that 50 unique color filters can be manufactured, that the sensor network is deployed from $2,400ft$ (necessary for the time-constrained relaxation) and that the time required for reporting connectivity grows linearly, with an initial reporting period of 160sec, as used in a real world tracking application [1]. The localization duration results, as presented in Table 1, are depicted in Figure 25.

As shown, for all practical purposes the time required by the space constrained relaxation techniques is 0sec. The same applies to the color constrained relaxation, for which the localization time is 0sec (if the number of colors is sufficient). Considering our assumptions, only for a network of size 50 the color constrained relaxation works. The localization duration for all other network sizes (100, 150 and 200) is infinite (i.e., unique color assignments to sensor nodes can not be made, since only 50 colors are unique), when only color constrained relaxation is used. Both the connectivity constrained and time constrained techniques increase linearly with the network size (for the time constrained, the Central device deploys sensor nodes one by one, recording an image after the time a sensor node is expected to reach the
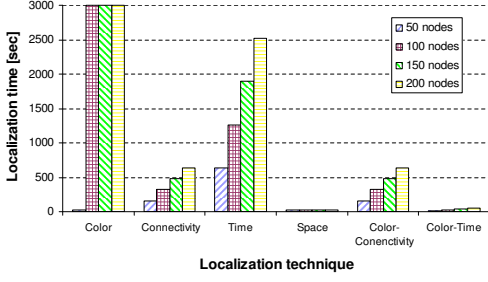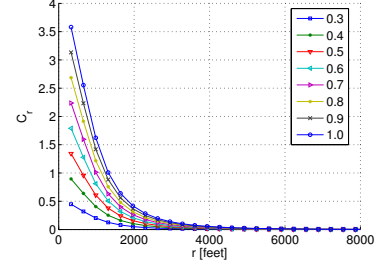
Figure 25. Localization time for different label relaxation schemes

Figure 26. Apparent contrast in a clear atmosphere

Figure 27. Apparent contrast in a hazing atmosphere

ground).

It is interesting to notice in Figure 25 the improvement in the localization time obtained by simply combining the color and the connectivity constrained techniques. The localization duration in this case is identical with the connectivity constrained technique.

The combination of color and time constrained relaxations is even more interesting. For a reasonable localization duration of 52seconds a perfect (i.e., $0ft$ localization error) localization system can be built. In this scenario, the set of sensor nodes is split in batches, with each batch having a set of unique colors. It would be very interesting to consider other scenarios, where the strength of the space constrained relaxation (0sec for any sensor network size) is used for improving the other proposed relaxation techniques. We leave the investigation and rigorous classification of such technique combination for future work.

5.5 System Range

In this section we evaluate the feasibility of the Star-Dust localization framework when considering the realities of light propagation through the atmosphere.

The main factor that determines the range of our system is light scattering, which redirects the luminance of the source into the medium (in essence equally affecting the luminosity of the target and of the background). Scattering limits the visibility range by reducing the apparent contrast between the target and its background (approaches zero, as the distance increases). The apparent contrast C_r is quantitatively expressed by the formula:

$$C_r = (N_r^t - N_r^b)/N_r^b \qquad (10)$$

where N_r^t and N_r^b are the apparent target radiance and apparent background radiance at distance r from the light source, respectively. The apparent radiance N_r^t of a target at a distance r from the light source, is given by:

$$N_r^t = N_a + \frac{I\rho_t e^{-2\sigma r}}{\pi r^2} \qquad (11)$$

where I is the intensity of the light source, ρ_t is the target reflectance, σ is the spectral attenuation coefficient ($\approx 0.12 km^{-1}$ and $\approx 0.60 km^{-1}$ for a clear and a hazy atmosphere, respectively) and N_a is the radiance of the atmospheric backscatter, and it can be expressed as follows:

$$N_a = \frac{G\sigma^2 I}{2\pi} \int_{0.02\sigma r}^{2\sigma r} \frac{e^{-x}}{x^2}\,dx \qquad (12)$$

where $G = 0.24$ is a backscatter gain. The apparent background radiance N_r^b is given by formulas similar with Equations 11 and 12, where only the target reflectance ρ_t is substituted with the background reflectance ρ_b. It is important to remark that when C_r reaches its lower limit, no increase in the source luminance or receiver sensitivity will increase the range of the system. From Equations 11 and 12 it can be observed that the parameter which can be controlled and can influence the range of the system is ρ_t, the target reflectance.

Figures 26 and 27 depict the apparent contrast C_r as a function of the distance r for a clear and for a hazy atmosphere, respectively. The apparent contrast is investigated for reflectance coefficients ρ_t ranging from 0.3 to 1.0 (perfect reflector). For a contrast C of at least 0.5, as it can be seen in Figure 26 a range of approximately $4,500 ft$ can be achieved if the atmosphere is clear. The performance dramatically deteriorates, when the atmospheric conditions are problematic. As shown in Figure 27 a range of up to $1,500 ft$ is achievable, when using highly reflective CCR components.

While our light source (3 million candlepower) was sufficient for a range of a few hundred feet, we remark that there exist commercially available light sources (20 million candlepower) or military (150 million candlepower [27]), powerful enough for ranges of a few thousand feet.

6 StarDust System Optimizations

In this section we describe extensions of the proposed architecture that can constitute future research directions.

6.1 Chained Constraint Primitives

In this paper we proposed four primitives for constraint-based relaxation algorithms: color, connectivity, time and space. To demonstrate the power that can be obtained by combining them, we proposed and evaluated one combination of such primitives: color and connectivity. An interesting research direction to pursue could be to chain more than two of these primitives. An example of such chain is: color, temporal, spatial and connectivity. Other research directions could be to use voting scheme for deciding which primitive to use or assign different weights to different relaxation algorithms.

6.2 Location Learning

If after several iterations of the algorithm, none of the label probabilities for a node n_i converges to a higher value, the confidence in our labeling of that node is relatively low. It would be interesting to associate with a node, more than one label (implicitly more than one location) and defer the label assignment decision until events are detected in the network (if the network was deployed for target tracking).

6.3 Localization in Rugged Environments

The initial driving force for the StarDust localization framework was to address the sensor node localization in extremely rugged environments. Canopies, dense vegetation, extremely obstructing environments pose significant challenges for sensor nodes localization. The hope, and our original idea, was to consider the time period between the aerial deployment and the time when the sensor node disappears under the canopy. By recording the last visible position of a sensor node (as seen from the aircraft) a reasonable estimate of the sensor node location can be obtained. This would require that sensor nodes posses self-righting capabilities, while in mid-air. Nevertheless, we remark on the suitability of our localization framework for rugged, non-line-of-sight environments.

7 Conclusions

StarDust solves the localization problem for aerial deployments where passiveness, low cost, small form factor and rapid localization are required. Results show that accuracy can be within $2ft$ and localization time within milliseconds. StarDust also shows robustness with respect to errors. We predict the influence the atmospheric conditions can have on the range of a system based on the StarDust framework, and show that hazy environments or daylight can pose significant challenges.

Most importantly, the properties of StarDust support the potential for even more accurate localization solutions as well as solutions for rugged, non-line-of-sight environments.

8 References

[1] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "An energy-efficient surveillance system using wireless sensor networks," in *MobiSys*, 2004.

[2] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based counter-sniper system," in *SenSys*, 2004.

[3] A. Arora, P. Dutta, and B. Bapat, "A line in the sand: A wireless sensor network for trage detection, classification and tracking," in *Computer Networks*, 2004.

[4] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *ACM SenSys*, 2004.

[5] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *ACM SenSys*, 2004.

[6] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Mobicom*, 2001.

[7] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Mobicom*, 2000.

[8] M. Broxton, J. Lifton, and J. Paradiso, "Localizing a sensor network via collaborative processing of global stimuli," in *EWSN*, 2005.

[9] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *IEEE Infocom*, 2000.

[10] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-assisted topology generation for auto-localization in sensor networks," in *IEEE Infocom*, 2005.

[11] P. N. Pathirana, A. Savkin, S. Jha, and N. Bulusu, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Transactions on Mobile Computing*, 2004.

[12] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *ICAASSP*, 2001.

[13] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, A. Nadas, K. Molnar, S. Dora, and A. Ledeczi, "Radio interferometric geolocation," in *ACM SenSys*, 2005.

[14] K. Whitehouse, A. Woo, C. Karlof, F. Jiang, and D. Culler, "The effects of ranging noise on multi-hop localization: An empirical study," in *IPSN*, 2005.

[15] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha, "Resilient localization for sensor networks in outdoor environment," UIUC, Tech. Rep., 2004.

[16] R. Stoleru and J. A. Stankovic, "Probability grid: A location estimation scheme for wireless sensor networks," in *SECON*, 2004.

[17] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, 2000.

[18] T. He, C. Huang, B. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free localization schemes in large scale sensor networks," in *ACM Mobicom*, 2003.

[19] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *IPSN*, 2003.

[20] D. Niculescu and B. Nath, "Ad-hoc positioning system," in *IEEE GLOBECOM*, 2001.

[21] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke, "A high-accuracy low-cost localization system for wireless sensor networks," in *ACM SenSys*, 2005.

[22] K. Römer, "The lighthouse location system for smart dust," in *ACM/USENIX MobiSys*, 2003.

[23] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE JRA*, 1987.

[24] C. L. Archer and M. Z. Jacobson, "Spatial and temporal distributions of U.S. winds and wind power at 80m derived from measurements," *Geophysical Research Jrnl.*, 2003.

[25] "Team for advanced flow simulation and modeling." [Online]. Available: http://www.mems.rice.edu/TAFSM/RES/

[26] K. Stein, R. Benney, T. Tezduyar, V. Kalro, and J. Leonard, "3-D computation of parachute fluid-structure interactions - performance and control," in *Aerodynamic Decelerator Systems Conference*, 1999.

[27] Headquarters Department of the Army, "Technical manual for searchlight infrared AN/GSS-14(V)1," 1982.