Incremental Multi-Robot Deployment for Line-of-Sight Chains Using Only Radio Signal Strength

John O'Hollaren and Dylan Shell

Abstract

Several interesting robotics applications involve maintaining or determining visual line-of-sight. This paper describes a method for determining whether robots are within line-of-sight of one another using only signal statistics gathered from wireless radios. We introduce a method that compares the signal strength statistics extracted from short windows of data against an idealized fading distribution. Using this method we have developed an algorithm for incrementally deploying a chain of robots whilst maintaining the line-of-sight constraint. We demonstrate and validate the approach on simple physical robots.

The effectiveness of certain important robotic capabilities, including communication and relative pose-estimation, are greatly diminished in obstacle-ridden environments. Walls, buildings, and other objects all impede data acquisition. One solution to this problem is to ensure that robots navigate around obstacles while collectively maintaining line-of-sight. Even when occlusion and attenuation effects are not entirely mitigated, some capabilities may be improved through use of *a priori* knowledge regarding a line-of-sight relationship between robots (*e.g.*, in wireless communication where one may apply models which are implausible more generally).

This paper describes simple and efficient methods for determining (and therefore maintaining) line-of-sight properties between robots requiring no external sensors. We outline four methods for achieving these ends using packet reception time, statistical outliers, and two fading models. The most successful of the methods uses wireless radio signal strength readings and an analysis of small-scale fading statistics to evaluate the degree to which collected data are explained by a Rayleigh fading model. We demonstrate that a sequence of signal-strength readings can provide adequate evidence for real-time estimation of the line-of-sight relationship between two wireless radios — and that an analogous method using the Ricean model is less reliable. Our evaluation of performance as a function of window size, shows that data collected over a few meters suffices for practical disambiguation.

We apply the developed Rayleigh-based method to the problem of deploying a chain of robots so as to maintain line-of-sight between robots. Inspired by Payton et al.'s work (**?**), we envision a system in which robots spread out to cover an environment whilst maintaining a constraint that ensures that any robot is visible from some other robot. This permits, for example, a human to be visually guided through an environment by moving from robot to robot, which is useful for applications like evacuation assistance (*cf.* **?**). Experiments with robot deployment within our office building show that, although the two factors of line-of-sight decision accuracy and data window size are coupled, the method may be used to practicably identify the line-of-sight boundary. As an illustration of minimalism, it is perhaps surprising that a visual line-of-sight capability can be provided for "free" when using hardware equipped with standard communications devices.

Related Work and Background

Received Signal Strength Indication (RSSI) at a receiver is a measure of the power that arrives at that receiver having propagated through the environment from a transmitting source. It is often stated that the relationship between radio received signal strength and distance is not a straightforward function, especially within indoor environments (**?**). If, however, one is told that the transmitter and receiver are connected by a line-of-sight connection, then a given signal strength reading provides evidence for a limited range of distances. Some authors (*e.g.*, **?**) assume that information about the line-of-sight property is provided as an input.

One naïve way to automatically detect the line-of-sight connection is by thresholding mean RSSI. Although easy to implement, this fails when any given value could be accounted for either by a short distance non line-of-sight connection with significant attenuation (*e.g.*, through walls) or a longer distance line-of-sight connection. Without additional assumptions about the environment, the two cases can not be disambiguated.

The Rayleigh-based method introduced in this paper focuses on the small-scale fading statistics rather than mean power or signal strength and consequently does not explicitly depend on thresholding signal strength values. We are aware of few papers that make use of small-scale fading effects for robotics. **?** exploit small-scale fading to sample a neighborhood of points in order to increase connectivity. They are able to derive an estimate of the number of samples necessary by assuming a Rayleigh model. We are unaware any other application of online modeling of small-scale fading effects for robotics applications.

Line-of-Sight Detection

A robot's environment impacts the path, strength, and timing of signals sent between it and other robots. We exploit these differences to classify robotic relationships into two distinct categories: line-of-sight or non line-of-sight. The experiments required two main sets of control data: one for each of the controlled line-of-sight conditions. Each main dataset included subsets encompassing different environmental features.

The first, 'line-of-sight,' dataset included data from two physical robots placed in a corridor approximately 10 meters apart. Each robot slowly executed a simple wall-following algorithm while recording the strength of an 802.11g Ad-Hoc network using an Atheros Communications AR5413 wireless card. During the runs, we ensured maintenance of the line-of-sight constraint throughout, or the run was thrown out. We performed similar steps to collect the second, 'non line-of-sight,' dataset.

Packet Reception Time

Initial investigation resulted in the following observation: average time between received packets serves as a simple, yet surprisingly accurate distinguishing factor between the two classes of data (line-of-sight and non line-of-sight). Figure 1 clearly illustrates this concept by showing the packet reception time difference for ten total runs: five line-of-sight runs and five non line-of-sight runs. It appears that using a threshold package reception time of approximately 0.087 seconds (the black dotted line), the datasets can be easily separated.

Attempting to exploit this fact and use it for online classification uncovered an important flaw. When the distance between robots varies considerably, determining if differences in packet time are due to the nature of the environment or simply alterations in distance between the robots becomes difficult. Although this method has limited applicability when there is significant variation in the transmitter-receiver distance, it is worthy of description because particular environments (*e.g.*, small buildings) reduce this variation and the method may be used even when RSSI readings cannot be obtained. This can be the case when only user-level access is granted to the network interface or when the hardware device does not support querying RSSI on a per packet basis.

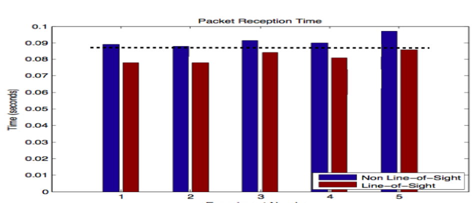

Figure 1: A comparison of average time between received packets in five control runs.

Motivating RSSI-based Detection: RSSI Extrema

Basic statistical analysis of the RSSI data uncovered an unexpected (and perhaps counter-intuitive) relationship between data outliers and the line-of-sight condition. Signal data between non line-of-sight robots showed a significant bias of outlying data points towards one side of the dataset. Similar signal readings from line-of-sight robots did not exhibit such a bias, as shown in Figure 2.

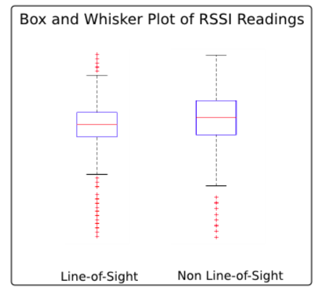

Figure 2: A Box and Whisker plot showing the uneven distribution of outliers in the non line-of-sight case.

Each red cross, or outlier, represents a data point that falls below 1.5 times the Interquartile Range (IQR) from the 25th percentile, Q(1), or above 1.5 times the IQR from the 75th percentile, Q(3). Equation 1 illustrates this rule. We were able to classify the robot's environment by counting the percentage of data points that qualified as outliers. Any dataset composed of less than 0.3 percent outliers was generally from a non line-of-sight run.

$$Outlier = \begin{cases} x : x < Q(1) - 1.5 * IQR \\ x : x > Q(3) + 1.5 * IQR \end{cases} \tag{1}$$

The uneven distribution shows that signal strength readings in the two cases lead to distinctly different patterns in data frequency; the probability distributions have a different shape. In fact, the non line-of-sight data with the long tail adheres reasonably to the Rayleigh fading model, which constitutes our final classification method.

The Rayleigh Fading Model

The Rayleigh fading model is a theoretical model of the statistical fading properties for a radio in a medium which receives a signal that has been sufficiently scattered to approach the receiver from all angles with equal probability (**?**). By invoking the central limit theorem, the received power is shown to be Rayleigh distributed according to Equation 2.

$$f(x;\sigma) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2} \tag{2}$$

The single parameter for the Rayleigh p.d.f., σ, is commonly modeled by its maximum likelihood estimate shown in Equation 3, given N independent and identically distributed Rayleigh random variables.

$$\hat{\sigma} = \sqrt{\frac{1}{2N}\sum_{i=0}^{N} x_i^2} \tag{3}$$

Our method takes advantage of this maximum likelihood estimate to create a single, constant value of σ that we utilize throughout the execution of our algorithm. This experimentally derived value for σ incorporates an average of values after normalization (described below) and from multiple runs in a variety of environments.[1]

In order to ensure that the classification method did not inadvertently exploit variations in the signal mean—which was observed to be a function of a particular environment and the exact range between transmitter and measuring receiver—the first step in the method involves normalization of the received data to fit between 0 and 1. Subsequent plots label this ”Normalized RSSI.”

Although clearly an idealization, when two robots are non line-of-sight, there exists no single dominant signal component between the robots. Instead, signals reflect off of various surfaces, resulting in a large number of different paths between the two robots. By the time signals reach the receiver, they are approaching from a variety of different directions, thus satisfying the requirement for Rayleigh fading. Figure 3 shows an example of RSSI data between two non line-of-sight robots compared against the Rayleigh fading model. Clearly the data matches the idealized Rayleigh distribution well.

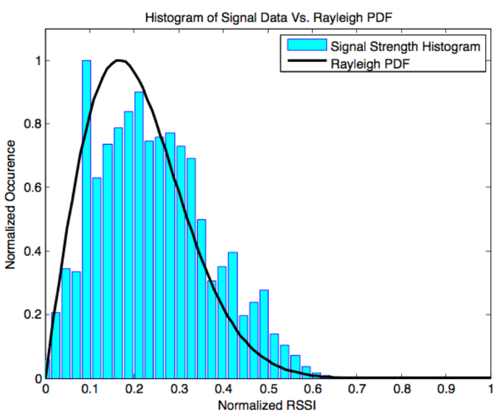

Figure 3: Rayleigh pdf imposed on non line-of-sight data.

On the other hand, signals between two line-of-sight robots travel on an unobstructed path from the emitted signal to the receiver. The resulting dominant direction of signal propagation violates the Rayleigh fading assumptions. As a result, the data do not match the distribution. Figure 4 shows the lack of correlation between the signal data and the Rayleigh fading model for two line-of-sight robots.

The Rayleigh-based Method

The difference in root mean squared error (RMSE) between the Rayleigh p.d.f. and the signal data in the line-of-sight and non line-of-sight cases can be used to make a prediction

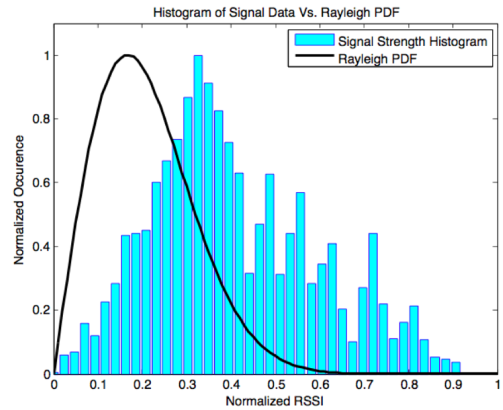

Figure 4: Rayleigh pdf and line-of-sight data.

of line-of-sight or non line-of-sight. A low RMSE value, which indicates a good fit, suggests a non line-of-sight relationship. A high RMSE value, indicating a poor fit, suggests a line-of-sight relationship.

Our method, shown in Figure 5, takes advantage of this fact through a multi-step process. Raw RSSI readings are converted to the power spectrum, filtered, normalized, and then compared with a Rayleigh p.d.f. for goodness of fit. The environment between the two robots can then be classified as either line-of-sight or non line-of-sight based on the value of the RMSE value relative to the designated RMSE threshold.

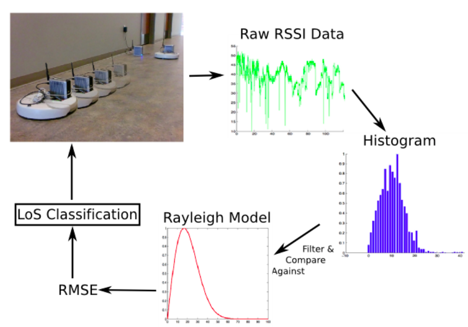

Figure 5: An overview of the Rayleigh-based classification method.

The method involves two important dependent parameters: the amount of data used for fit and the RMSE threshold used convert the quality of fit into a classification. Since the algorithm's running time scales (linearly) with the number of RSSI data, using few RSSI readings leads to a rapid classification. However, the quality of the classification deteriorates with too few samples because the histogram does not reflect the underlying distribution adequately and the signal fluctuations exacerbate the situation. On the other hand, relying on many samples limits the applicability of the method: it can take a long time to collect and process the data. In collecting a large sample of data, the robot is likely to have moved a significant distance and will have

[1]We collected data across several environment types and multiple buildings to arrive at a value of $\sigma = 0.5$. It is probable that finding a particular σ for a given environment would improve the quality of fit for those data, at the loss of generality.

passed through conditions of both line-of-sight and non line-of-sight during that time. The ideal sample size lies somewhere in between the two extremes: small enough to give a quick result reflecting recent positions but large enough to ensure accuracy. This ideal value can be discerned by viewing a graph of sample size versus RMSE, as shown for a non line-of-sight case in Figure 6. From this graph, it is clear that we can use a sample size between 2500 and 3000 with a RMSE threshold of 0.28.

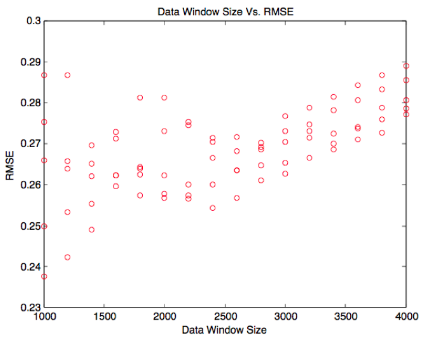

Figure 6: Plot showing RMSE versus window size for two non line-of-sight robots.

The Analogous Rician-Based Method

The argument made above regarding the quality of fit of non line-of-sight data to the Rayleigh model suggests that a similar approach be employed to fit line-of-sight data. The explanation highlights the expectation that line-of-sight connections do not have radially symmetric received power. If the line-of-sight connection has a single dominant signal component, then the Ricean Fading model is generally regarded as applicable (**?**). The Rician p.d.f. is shown in Equation 4, where J_0 is the modified Bessel function of the first kind and zero order.

$$f(x;\sigma,\mu) = \frac{x}{\sigma^2} * e^{fracx^2+\mu^2-2*\sigma^2} * J_0(\frac{x\mu}{\sigma^2}) \quad (4)$$

Instead of associating the signal data between non line-of-sight robots with the Rayleigh p.d.f., one might compare signal data between line-of-sight robots with the Rician p.d.f.. Our findings were that the quality of fit between the received data and the Rician model was not a particularly useful statistic because the optimal Rician p.d.f.'s for the line-of-sight and non line-of-sight cases were very similar. As evidenced by Figure 7, the approximate parameters which resulted in the best fit for the typical line-of-sight case (σ=1, μ=2) were very near to those which were optimal for the non line-of-sight case (σ=1, μ=1.5). As a result, there was no clear RMSE threshold as there was for the Rayleigh model, making the Rician method a poor distinguisher of the robot's environment. We hypothesize that the complexity of indoor environments provides multiple directions of increased received power (*e.g.,* reflections down corridors result in non-trivial regularity in angular power distribution) which violates the idealized representation for Rician fading.

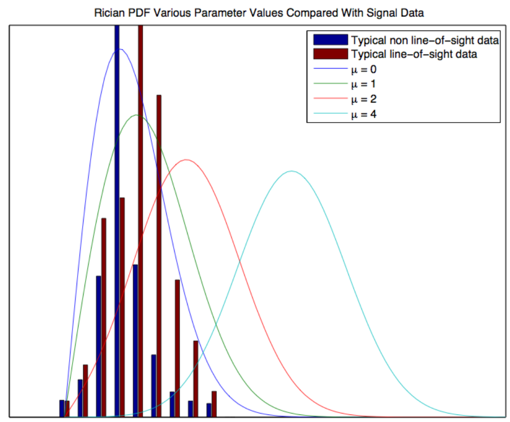

Figure 7: Comparison of normalized Rician p.d.f.s against Normalized RSSI.

Implementation: Robotic Deployment

Line-of-Sight Boundary Detection

We constructed controlled scenarios in which a moving robot maintained either line-of-sight or non line-of-sight throughout its entire motion. Figure 8 illustrates the results of running the recorded data from these runs through the previously outlined Rayleigh-based method. The method output only five false readings for fifty-one inputs resulting in an accuracy rate of 90.2 percent.

	True LoS	True non-LoS
Predicted LoS	12	1
Predicted non-LoS	2	14

	True LoS	True non-LoS
Predicted LoS	6	1
Predicted non-LoS	1	14

Figure 8: Confusion matrices for two robots analyzing their environment using the proposed method. The matrix on the left is for marble hallway, the matrix on the right is for a concrete hallway.

Deployment Algorithm

In order to further evaluate the method through the creation of line-of-sight chains, we developed an incremental deployment algorithm for a team of robots. The algorithm utilizes the fact that non line-of-sight is detected as the robots round corners to allow each robot to move one corner, or line-of-sight boundary, further into the environment than the previous robot.

Deployment Experiment: Two Robots

The simplest scenario where this technique can be applied is that of two robots: one stationary base station and one receiver. The lone moving robot moves out until it detects non line-of-sight, attempts to move back to line-of-sight with the base station, and then stops. We ran this scenario multiple times down the same 'L' shaped hallway with the stationary host located six and a half meters from the corner. The results for a series of five runs, in Figure 9, show that the robots reliably found the line-of-sight boundary (the corner).

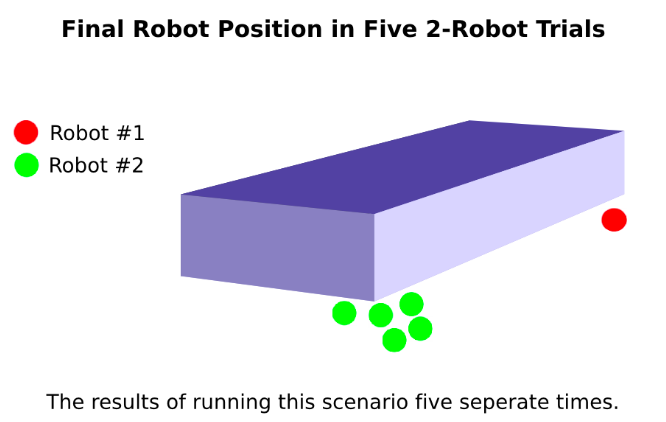

Figure 9: The results of five separate two robot trials show a satisfactory degree of accuracy; the moving robot is able to consistently stop very near to the line-of-sight boundary.

Three Robot Chain

In three agent demonstrations, the method showed that while robots consistently maintain the line-of-sight constraint, they do not necessarily do so with maximum spread. Figure 10 depicts the results of five consecutive three-agent trials.

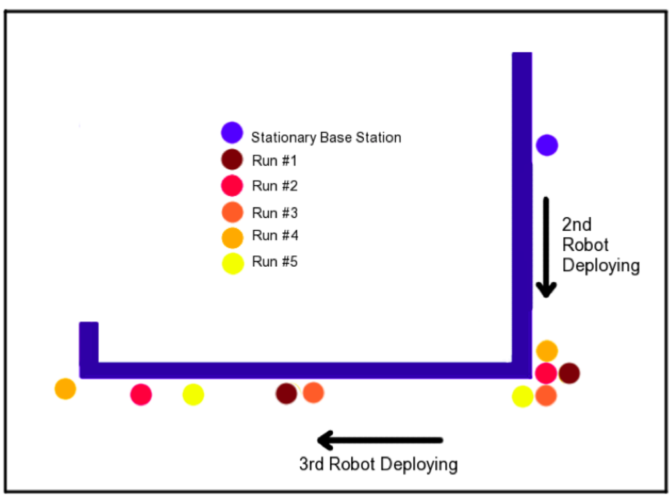

Figure 10: The final locations of the robots in three agent trials.

The method still creates line-of-sight chains, as evidenced in the figure, albeit not always a maximally efficient one. If the second robot stops even a little short of the boundary, subsequent robots do not have a reliable reference to ping. As a result, they will detect non line-of-sight early and stop short of the next boundary, which explains the non-maximal line-of-sight chain.

Limitations and Future Work

The primary limitation in line-of-sight determination through exploitation of wireless communication devices is that the environment must consist of obstacles that have significant attenuational properties.

Our experience showed that the method was successful when employed around objects with significant attenuational properties, but failed to disambiguate the two cases in an open space with thin movable office partitions. In principle, more accurate measurement of signal reading could address this issue, but off-the-shelf hardware is not likely to suffice. A more significant limitation stems from utilizing the method in two radically different environments. The parameters that work well for a large open area, for example, will not work well in a closed corridor.

One key research topic left untouched is the creation of a non-incremental deployment technique. This would involve each robot simultaneously fanning out from a central node and creating a line-of-sight zone, instead of a line-of-sight chain as detailed in this paper.

Conclusion

This paper demonstrates the feasibility of using communication devices to disambiguate connections between robots that are line-of-sight from those that are non line-of-sight. We describe four simple, efficient methods for conducting this determination (in increasing complexity): (1) A packet-reception time method; (2) A statistical outlier method; (3) A Ricean-based method; (4) A Rayleigh-based method. We outline the limitations of the first and second methods, show that the third method appears to be ineffective despite its comparative sophistication, and motivate and illustrate the fourth method.

The Rayleigh-based method results in few false positives. The resulting accuracy allows the method to be applied to line-of-sight detection and multi-robot deployment schemes. In conclusion, this method demonstrates the use of a simple sensor reading to accomplish a complex task with several real-world applications.