# LIDAR Uncertainty Analysis and Scan Optimization

Colin Snow

May 2021

## 1 Executive Summary

An optimization routine is proposed which informs the process of collecting LIDAR data to reduce scanning time and increase accuracy. This process relies on creating simulated measurements with Monte Carlo sampling based on estimated positional data. The algorithm minimizes total positional uncertainty by choosing the optimal scanning position in each iteration, providing the maximum amount of information possible in each scan. If this method can be applied to drone-based LIDAR it could significantly reduce the time needed to scan objects while creating more detailed maps and requiring less pilot decision making. This could have important implications when LIDAR is used to gather information for disaster response or other time-sensitive applications, potentially saving lives while reducing sampling cost.

## 2 Background

LIDAR mapping has become extremely popular in the past several years due to a dramatic decrease in size, weight, and cost of sensors. LIDAR devices can now be mounted to drones or planes or simply carried around by people to create precise maps of an area without the need for the laborious process of traditional surveying. Drone-based LIDAR has started being used in more time critical applications such as search and rescue and disaster mitigation, adding new demand for faster, more efficient scans. In cases like these the time to scan an area and the accuracy of that scan can have real life or death consequences.

While many very good algorithms exist for patching together LIDAR scans and optimizing maps, little has been done to optimize the actual scanning process with respect to the inherent inaccuracies of the sensing equipment. Because 3D models are usually generated in xyz Cartesian space and LIDAR scans must be taken in spherical space, a conversion must be done between the two to convert a point in (r, theta, phi) to (x, y, z). While this conversion is mathematically trivial for a single point, any error in the position of that point will be propagated into the Cartesian model. Typically, LIDAR scanners have ratings for distance and angular accuracies to account for both the tolerance of the laser range finder and the angular motion equipment. Because these systems largely function independently, their errors can be assumed to be independent. An error in distance does not affect the error in angle and vice versa. However, when we try to transform these errors into Cartesian coordinates, we find that the transformed distributions are correlated, and that this correlation depends not only on the errors in measurement but also on the measured position. The magnitude of error tends to grow with the measured distance and each component of error grows with the distance in that axis.

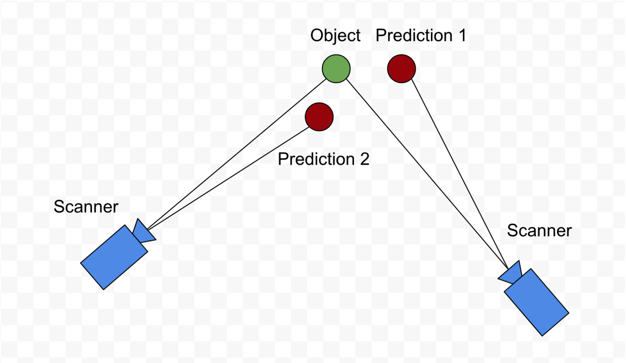

Figure 1: Two LIDAR scanners taking a scan of the same object. If the scanners were perfectly precise they would find the object to be at the same position. In reality, their errors lead to to different predictions.

This inherent correlation in x,y,z is often left unaccounted for in analyses which include LIDAR data such as SLAM and other mapping algorithms either because it adds computational complexity that can cause divergence or is simply not well understood. In cases where the primary objective is to determine the position of the observer it may be possible to throw away these correlations and still get a good enough result, but for precise mapping it is critical to understand how this error propagates.

This understanding of the inherent uncertainty of LIDAR can be used to quantify the quality of scans and, critically, to prescribe how measurements can be taken to get the greatest increase in quality in the fewest possible measurements. This information can then be used to instruct pilots and operators on where to take new scans and when they have reached a desired level of quality and can move on to a new area, taking the guesswork out of scanning and allowing larger areas to be scanned in the same amount of time.

# 3 Formulation

Suppose we have some set of objects we would like to know the position of and some set of scans that have already been performed of these objects from different points around them. We would like to know where to take the next scan in order to decrease our uncertainty about the positions of the points as much as possible. This is quite a complex and abstract problem statement so we will walk through the various pieces and then tie it all together.

## 3.1 Assumptions

Before we even begin to solve this problem we must understand the assumptions we are making in our model. For the sake of simplicity, we will work in 2D space, though the same concepts should apply directly to 3D or even higher dimensions. We also assume that all objects in the space are distinct point objects (no surfaces) and that we know which object we are looking at and can not get them confused. This is clearly not true in real applications but object recognition algorithms and other simplifications exist which could turn real data into a model of the form we consider. We also assume that the observation position is known with certainty to reduce the number of error terms but uncertainty in observation location could be implemented. Finally, we assume that range and angular error are uncorrelated, normally distributed random variables. This assumption is probably a relatively good model of sensor error but to get the best results it would be advantageous to characterize the error of a particular sensing suite. The analysis does not depend on the distribution of these values or their independence, and would work just as well with a different distribution or level of correlation.

## 3.2 Quantifying Uncertainty

### 3.2.1 LIDAR Error

Imagine there is an object in space and that we want to find its position. Before we take any measurements its uncertainty is effectively infinite. We have no idea where it is. Through the process of taking measurements of it we gain information which can be used to estimate its position. However, these measurements each have some error associated with them. We do not know the true error because we do not know where the object actually is, but we can construct a model of how much error we might expect to have in our measurement based on the type of measurement we take and information about our measuring tools.

Suppose that we have a LIDAR scanner and want to take a measurement from the origin (0,0) of a point at (1,0). When we take this measurement we assume that there will be some error associated with it and that this error will depend on both the distance and angular errors of our sensor. We can use a Monte Carlo evaluation to run this measurement many times and see the resulting distribution of points.

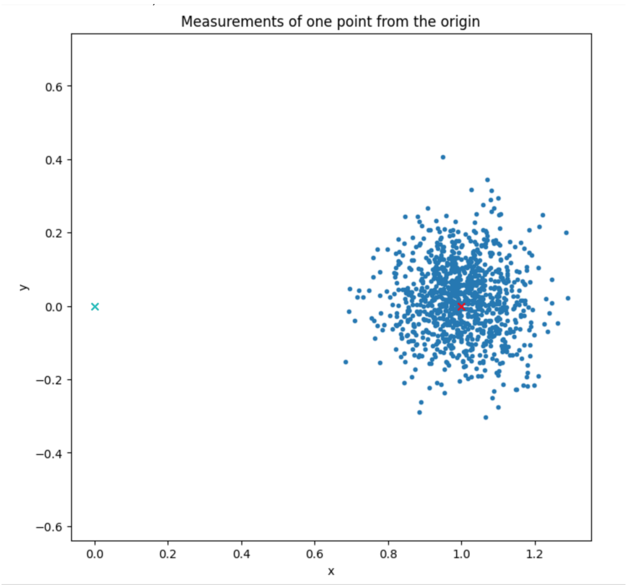

Figure 2: Measurements of a point at (1,0) from the origin. The blue x is the observer and the red x is the true object position. The measured values form a distribution around the true value.

As we can see, the sampled measurements form a cluster around the true value with some variance in both x and y. The size of this variance is determined by the inherent error in our measurement apparatus and how this error gets transformed when we go from polar to Cartesian spaces. Suppose instead that we want to take the same measurement but of three objects at (1,0), (2,0), and (3,0).

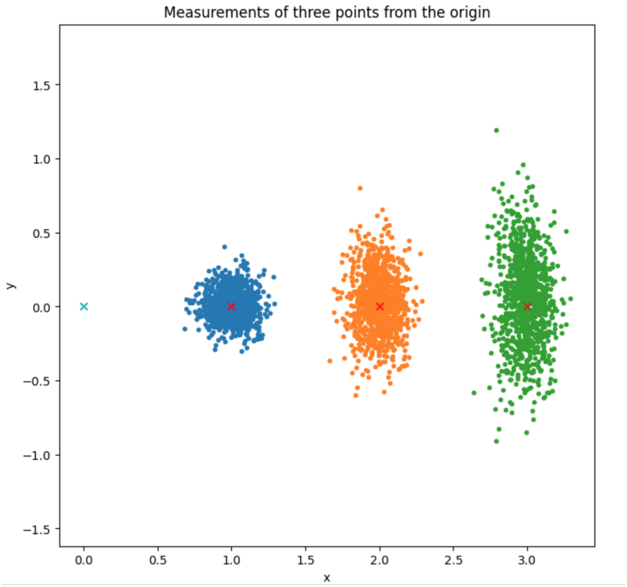

Figure 3: Measurements of three points form the origin. The blue x is the observer position and the red x's show the true positions of the objects. The error in position increases with distance because angular error has more effect on Cartesian error as the radius increases.

What we see is that the farther away points seem to have greatly exaggerated errors in y but relatively similar errors in x. If our measurement apparatus did not change between these experiments there must therefore be some geometric reason for this change. This once again goes back to the conversion between polar and Cartesian spaces and the propagation of error between the two. In 2 dimensions, a point $(r, \theta)$ can be converted to $(x, y)$ with the equations:

$$x = r * \cos\theta$$

$$y = r * \sin\theta$$

Because of this relationship, errors in r generally correspond proportionally to errors in $(x, y)$, while errors in $\theta$ tend to be magnified as r increases. This means that the distance we take our measurement from has little to no effect on the error in the direction we are looking but a large effect on error perpendicular to the observation direction. This accounts to what qualitatively looks like the vertical stretching of the distribution we see in figure 3.

### 3.2.2 Quantifying Error

We have now seen qualitatively what error distributions tend to look like for LIDAR data but to perform optimization we must find some quantitative way to measure uncertainty. The simplest method for quantifying uncertainty is to take the standard deviation of our data and use the x and y components of this to quantify error in their respective directions. Because we are concerned about minimizing absolute uncertainty, we will define the uncertainty of a point as the sum of the components of its standard deviation. This is a modelling choice which may not be reasonable in some circumstances so it is important to determine if assuming this would have an adverse affect for a particular application. Given this definition for a single point we can define total uncertainty as the sum of uncertainties over all observed points. This is the quantity we will attempt to minimize in our analysis. However, when choosing to take a measurement of a point from a new location, we do not want to include the new measurement if it would make us less certain about the true position. Specifically, for any given point if adding the new measurement will increase the sum of standard deviations for that point we will consider that measurement to not improve our accuracy. If this is the case we will simply exclude that measurement from the set of measurements of the point and take the sum of all the others.

### 3.2.3 Generating points from uncertain data

All the examples above have assumed that we could take some large quantity of samples in order to build a distribution. In a real application, of course, only one point is observed. Clearly, then, we need some way of simulating new observations given the data we have. To do this, we will assume that the point we measure is the true mean and construct a distribution about that point using Monte Carlo and the distributions for the error variables. By doing this, we effectively reverse the problem. Instead of creating a distribution of where new measurements might land given the true position, we create a distribution of where the true position might be given the measurements we have. Say, for example, we repeat the experiment in figure 3 but build the distributions around one of the measured points instead of the mean.

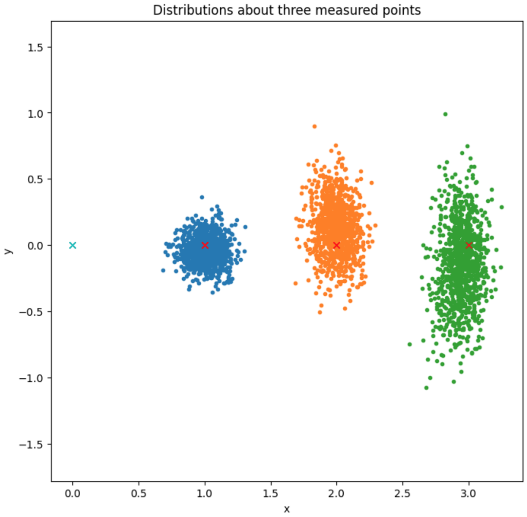

Figure 4: A repeat of the last experiment with distributions centered around a random observation instead of the true mean. In this case the distributions tend to be more spread when uncertainty about the true mean is large.

We see that the distributions still look quite similar but that they are offset by some amount of distance and angular error. Notice that the true mean (red x) tends to fall within the distributions and that as they get bigger the mean tends to fall farther from the center.

Unlike the previous process which required many observations, here we can take a single observation and use it to create many new samples programmatically. These new samples can then be used to estimate the uncertainty of the mean and thus quantify how uncertain we are about the position of the object.

### 3.2.4 Combining distributions

This method works well for a single observation of an object but when we start taking multiple observations we must decide how to combine them together. Surprisingly, this process can be made very simple with Monte Carlo. We can create a set of estimated points for each observation, combine these sets, and use the result to analyze uncertainty. This effectively overlays the distributions we have seen previously to create a distribution which has more information about the object.

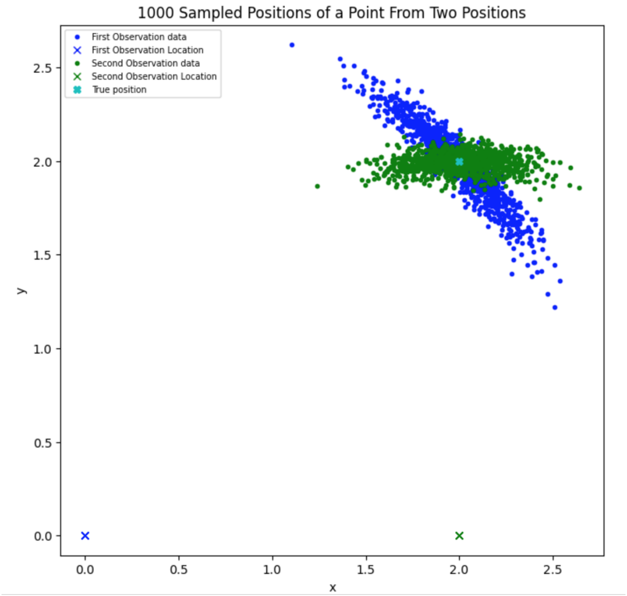

Figure 5: Samples of a point from two observation locations. The area where the blue and green distributions overlap has the highest density of points and is therefore the most likely point for the true mean to occur.

In figure 5 you can see the distributions which result from taking an observation from two different points. The addition of the two distributions creates an area of very dense points where they overlap. By taking the mean of this combined data we can estimate the true mean and by taking the standard deviation we can estimate our uncertainty about that mean.

## 3.3 Optimization

Now that we have a process for taking measurements, creating simulated data, and assessing the uncertainty of our observations, we can begin to frame an optimization problem to decide where scans should be taken.

Suppose we have some set of points at positions in $(x, y)$, each of which have been previously measured to be at some position $(x_{observed}, y_{observed})$ by a scanner at a position $(x_{observer}, y_{observer})$. Using the method defined previously we can create an error term $E$ which represents the sum of all our positional uncertainty given the measurements we have taken. We would like to find some position $(\hat{x}_{observer}, \hat{y}_{observer})$ which minimizes $E$ over all the points in the space. Specifically, we would like to solve the following problem:

$$minE(x, y)$$

$$wrt.x, y$$

While this problem does rely on the previous measurements taken they are not explicitly a parameter or constraint in optimization. They serve as additional information which is built upon to find a solution. The optimal point can then be found with a simple two-variable minimization routine.

# 4 Results

This optimization formulation very effectively finds the optimal point to take a new measurement to minimize uncertainty. The formulation of the problem in this way tends to lead to smooth surfaces of the objective with well defined minima.

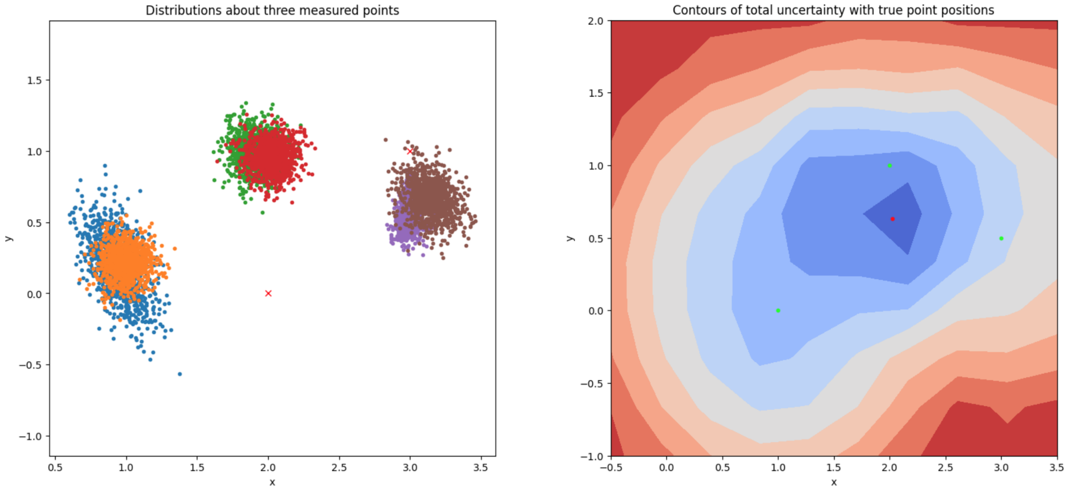

Figure 6: Left: Distributions about a set of three points with measurements taken from the red x spots. Right: The same points in green with contours of the objective overlaid and the optimal scanning position in red.

A sample problem is shown in figure 6 for a set of three points, each of which have two previous observations. The left side shows samples generated around the measured values and the right side shows contours of the objective versus the true position of the points. The optimization yields a value of (2.02, .632) as the minimum of error which is near the middle of the lowest contour. Intuitively, a value here makes sense. The chosen point is well centered between the objects and slightly closer to the two right ones. Understanding the shape of the objective can be a little easier in 3D, as shown in figure 7.

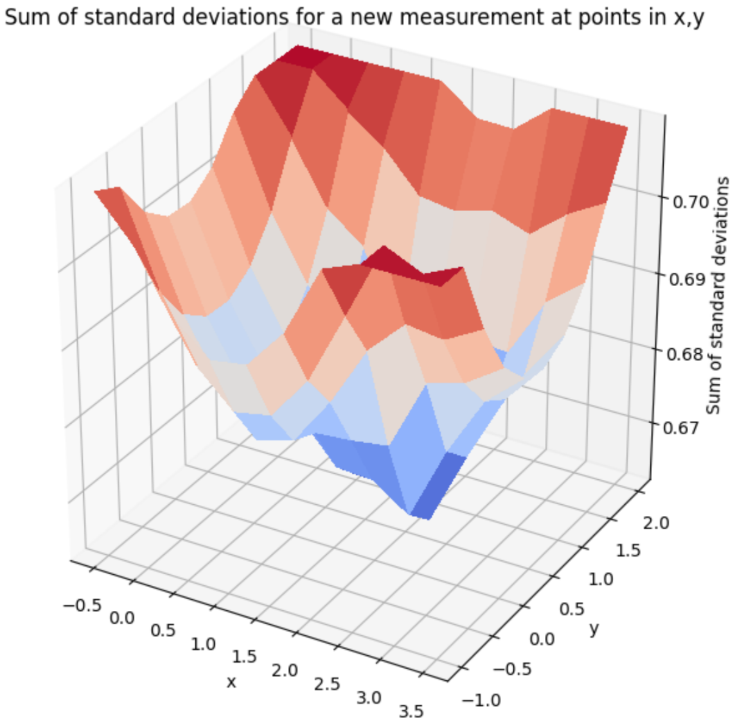

Figure 7: A 3D graph of E for swept values in x,y. Two local minima are present which represent the two best places to scan, with one being slightly better than the other.

Here you can see that there are actually two minima, one a bit closer to the left point and one a bit further away. Local minima are a relatively common occurrence in this formulation so it is important to use multiple restarts to ensure the optimization does not terminate prematurely. By adding the results of the optimal scan to the dataset and running the optimization again, a progressively better model of the objects can be developed. This can be done until an acceptable amount of uncertainty is reached.

After running many experiments with this method a few general trends seem to emerge. When no previous observations have been taken the algorithm tends to choose a point which is about centered between the objects. This makes sense because without any more information error is directly proportional to distance. Once one scan is taken the decisions get more complicated. If previous scans have not been taken near the center of the objects that point seems to still be a good option. However, if that region has been scanned already it will tend to pick positions which are either very close to the object with the most uncertainty or maximize the perpendicularity of the measurement to the previous measurements. This represents a trade off between eliminating a large quantity of error from one point or a small amount of error from many points. Interestingly, when the number of observations gets very large it tends to revert back to choosing the center of the objects in order to squeeze out any remaining error. In a real application the process would likely be stopped when this occurs as the improvement in the error gets very small.

## 5 Conclusion

This process of optimization could be used to increase the accuracy of LIDAR scans while reducing scanning time, increasing the effectiveness of LIDAR devices for use in lifesaving applications.

This method would take pressure off of drone pilots to simultaneously fly and ensure proper scan coverage while reducing total flying time.

While the method described here is purely theoretical and makes many simplifying assumptions for the sake of clarity I believe that an adaptation of this methodology would provide similarly advantageous results in a real-world setting. If this proves true it could significantly increase the usefulness of LIDAR devices and make them even more effective tools for emergency response or anything that needs to gather spatial information efficiently.

To this end it is important to consider the context in which this technology would be used. Disaster relief workers need technology which is easy to use and incredibly reliable. Any equipment failures or measurement errors seriously impede progress and can put lives at risk. The need for reliability informed many of the core assumptions and generalizations of this project. While choices like simplifying to 2D or assuming a known observer position do not substantially reduce generalizability, assumptions about measurement error and correlation can cause serious problems when switching between platforms or in unusual situations. These last two assumptions, which are common among existing solutions, mean that much of the information needed to assess reliability goes unrecorded. In cases where accuracy is critical, such as when assessing the structural stability of buildings, knowing the expected error bounds on readings can tell emergency personnel whether they have enough information to determine if the situation is safe and how to collect more information if they do not.

In a real world application of this methodology those same assumptions would have to be carefully considered in order to maximize usefulness for this application. At a minimum an implementation of this method would need to be generalizable to several sensor types, run efficiently on limited hardware so it could be used anywhere, and incorporate an intuitive interface so that it could be used by people without having to know how it works. While significant additional work would need to be done to determine all factors of a successful implementation, these three solve some of the most immediate and critical errors in existing implementations.