Galaxy aerial laser scanner has 500khz frequency so every 2 micros second it shoots a transmit laserbeam to the earth. However, the time to receive back scattered laser light of each transmit depends on its range (the distance from sensor to the earth).

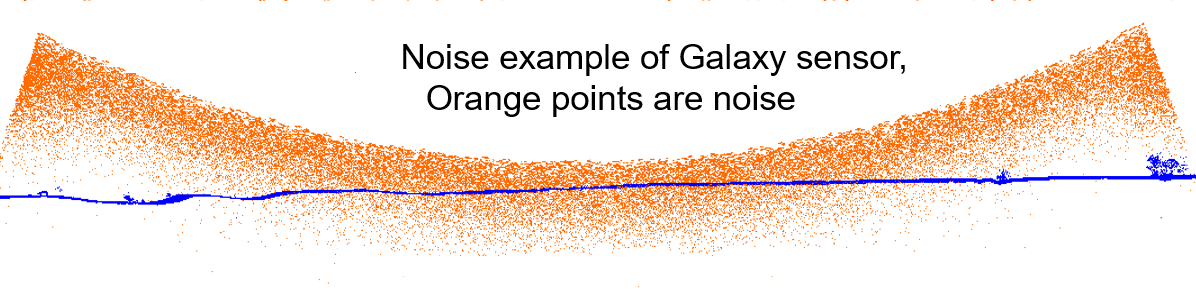
Based on this simple equation ‘x= vt’ we can calculate the range which the return light will be received from the same position that its correspondence laser beam hit the earth for this .

As is the distance than the beam goes and returns, the range will be half of that so:

|  |
| --- |
|  |
|  |

So, in distances for the ranges more than 300 m we are not capable of distinguish which back scattered is related to which transmit beam. For example, if a beam meets the earth with the range of 700 it takes 4.6 micro second to receive its back scattered by the sensor. During this time two next beams meet the earth and discrimination among corresponding lights will be difficult.

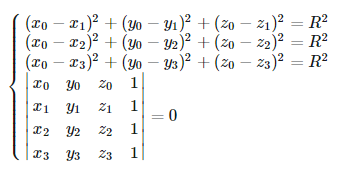
Moreover, since the laser beam at high flight attitude hit some articles in the air and close to sensor and receive their back scatters as 3D points which are called atmospheric noise.



To address this issue we wish to define different circles and calculate perpendicular distance of each point from the circle using Mestimator and then apply Maximum Likelihood to assess how this model represent points well.

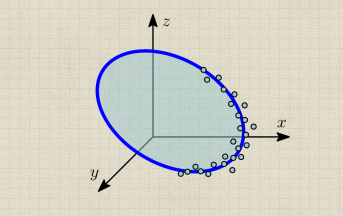
For this purpose, we can assume the slat range as a radius of a circle which its origin is located on the sensor. Then as we can define a circle trough each 3 non collinear points we have two different scenarios:

1. We can wire the geometry model for circles with given 3D coordinates of the points and the coordinate of origin and the radius are unknown. So the number of unknowns are 4 and the number of equations are the same as the number of circles. Since we have more equations than unknowns, we need to find an approximate solution by method of least-squares in 3D space. In fact, we have to solve the following system:



1. However, based on some articles as a circle in 3D is parameterized by six numbers: two for the orientation of its unit normal vector, one for the radius, and three for the circle center; it would be better to project 3D points to 2D and fit a circle in 2D and then project back to 3D. For instance, we can first using SVD decomposition found a plane that fits to the set of 3D points. Then, the 3D points will be projected onto the plane and new planar coordination for them will be got. Finally, the method of least-squares to fit a 2D circle into the planar points will be used and then the 2D fitting circle project back to the 3D coordination.

Here we describe a method how to fit a circle to the cluster of points in 3D space in detail. It utilizes the *singular value decomposition (SVD)* and the method of *least-squares* for the optimal circle fitting.



1. Introduction

Assume we have a set of n points P0,…,Pn−1 where Pi=(xi,yi,zi)T∈ . We want to find a circle that fits as close as possible to the set of points. A circle in 3D space can be represented by a parametric equation:

Pcircle(t)=rcos(t)u+rsin(t)(n×u)+C, 0≤t≤2π

With radius r, center C and normal unit vector n. Vector u is any unit vector perpendicular to n. If we specify orientation of the circle in space by angle ϕ and θ, we get

1. Algorithm

The circle fitting method can be split into the following steps:

1. Using SVD (Singular Value Decomposition) find the best fitting plane to the set of mean-centered points.
2. Project the mean-centered points onto the fitting plane in new 2D coords.
3. Using method of least-squares fit a circle in the 2D coords and get circle center and radius.
4. Transform the circle center back to 3D coords. Now the fitting circle is specified by its center, radius and normal vector.

#### Fitting plane by SVD

Assume we want to find a plane that fits as close as possible to the set of 3D points, and the closeness is measured by the square sum of orthogonal distances between the plane and the points. Lets introduce n×3 matrix of mean-centered points A=, where c=. Then the problem of finding the normal unit vector n for the fitting plane can be formulated as

Lets use singular value decomposition A=, where U and V are unitary matrices (orthonormal columns and rows), and Σ is a diagonal matrix containing singular values  ≥ ≥ ≥0. Using the SVD decomposition we can write

== = ++

where we introduced substitution b=. Thus, our expression is obviously minimized by choosing b=since  is the lowest singular value. We can finally find n by solving

  =b, which is trivial since V is an unitary matrix, so V=I.

n = = V = V(:,3)

Now we can see the expression  is minimized if and only if the normal vector n is chosen as the 3rd column of matrix V.

#### Projecting points onto the fitting plane

We can utilize the Rodrigues rotation formula to project 3D points on to the fitting plane and get their 2D X-Y coordinates in the coordinate system of the plane. we need to choose axis of rotation k as cross product between plane normal and normal of the new X-Y coordinates. Thus, k=n× .

Prot = Pcos(θ)+(k×P)sin(θ)+k⟨k,P⟩(1−cos(θ))

#### Circle fitting in projected 2D coordinates

Assume we want to fit a circle to set of n points. The implicit equation in 2D for a circle with radius r and center   can be arranged as

+ =

+=

+=

where we introduced the vector c = for unknown parameters. Applied to all input points, it yields to a system of linear equations

Ac = b

Where

Since we have more equations than unknowns, we need to find an approximate solution by method of least-squares, which minimizes the square sum of residuals , thus

C = min