Moving sistem

To make this sistem autonome we had the following problems

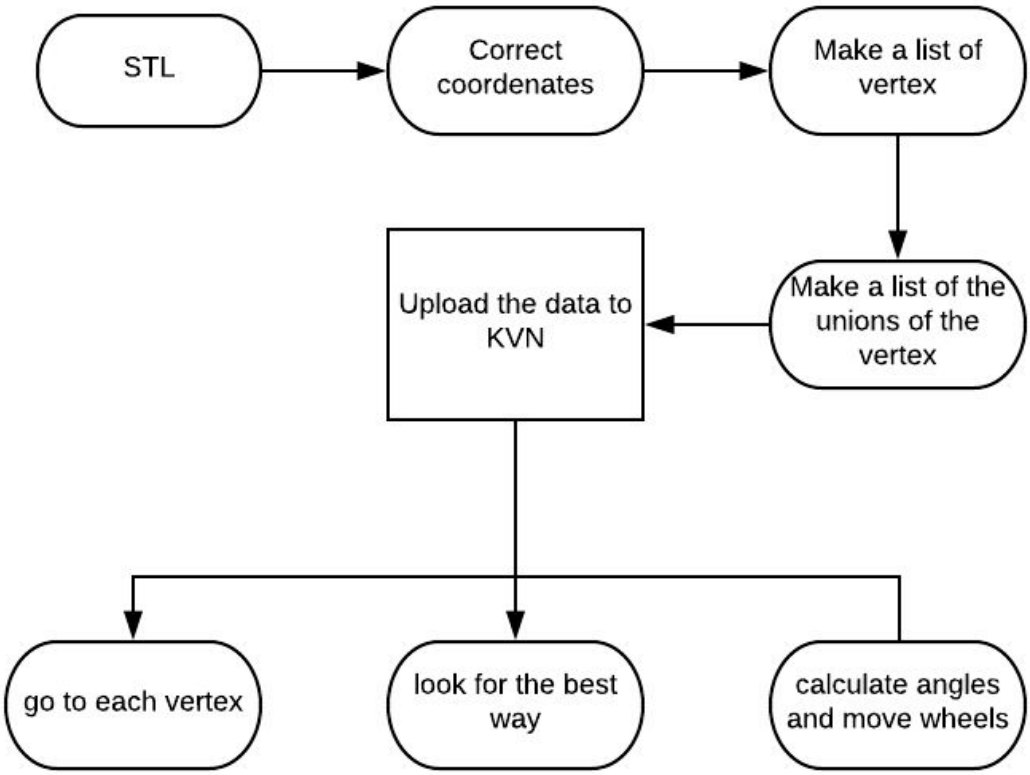
1. KVN has a 2D movement but it has to move in a 3D space
2. KVN has to be sure of move around the entire ship

Solution

The solution we propose is based on STL models. First we need the 3D model of the satellite/ spaceship , is not difficult to obtain because for design this vehicules the ingeniers makes an model by CAD software.

We need an app that transform the STL into a new format we call KVN .in this format we will have exactly the same information but in a more easy way to be read by the drone.

The next steps will be following by the program



STL

We chose this format to begin because is a very well-know format because is the one that uses 3D printers. This format transform the models to lots of triangles.

facet normal n_i n_j n_k
  outer loop
    vertex $v1_x$ $v1_y$ $v1_z$
    vertex $v2_x$ $v2_y$ $v2_z$
    vertex $v3_x$ $v3_y$ $v3_z$
  endloop
Endfacet

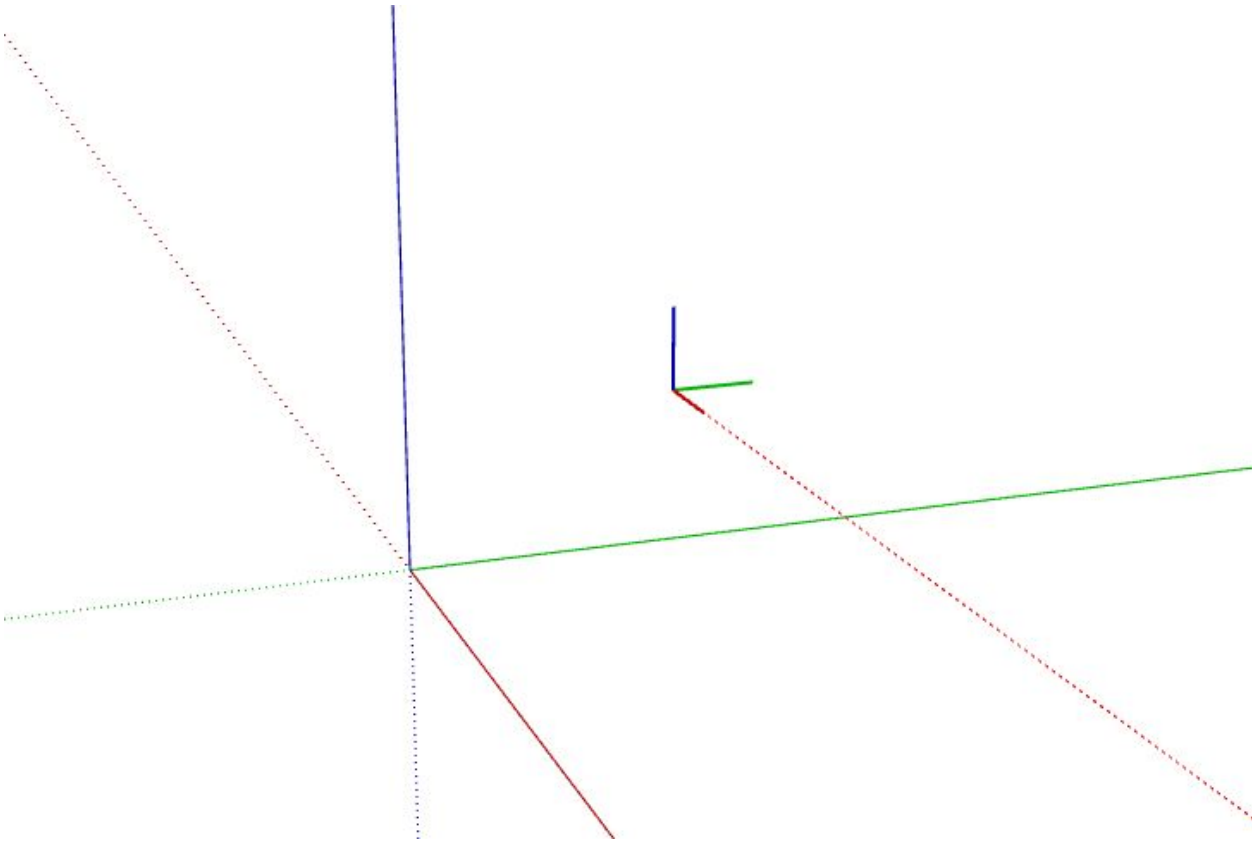
This is a real example of STL (ASCII) of a cube.

```
solid Example
   facet normal 0.000000e+00 0.000000e+00 -1.000000e+00
      outer loop
         vertex 0.000000e+00 1.000000e-02 0.000000e+00
         vertex 1.000000e-02 1.000000e-02 0.000000e+00
         vertex 0.000000e+00 0.000000e+00 0.000000e+00
      endloop
   endfacet
   facet normal 0.000000e+00 -0.000000e+00 -1.000000e+00
      outer loop
         vertex 1.000000e-02 1.000000e-02 0.000000e+00
         vertex 1.000000e-02 0.000000e+00 0.000000e+00
         vertex 0.000000e+00 0.000000e+00 0.000000e+00
      endloop
   endfacet
   facet normal -1.000000e+00 0.000000e+00 0.000000e+00
      outer loop
         vertex 0.000000e+00 0.000000e+00 1.000000e-02
         vertex 0.000000e+00 1.000000e-02 0.000000e+00
         vertex 0.000000e+00 0.000000e+00 0.000000e+00
      endloop
   endfacet
   facet normal -0.000000e+00 -1.000000e+00 0.000000e+00
      outer loop
         vertex 1.000000e-02 0.000000e+00 1.000000e-02
         vertex 0.000000e+00 0.000000e+00 1.000000e-02
         vertex 0.000000e+00 0.000000e+00 0.000000e+00
      endloop
```

From this data we can make a map of satellites or spaceships.

Correct coordinates

The model have their own coordinates but the position sistem has another coordinates, so we have to refer the coordinates to the position sistem.



$$(Xi, Yi, Zi) = (Xm, Ym, Zm) - (Xa, Ya, Za)$$

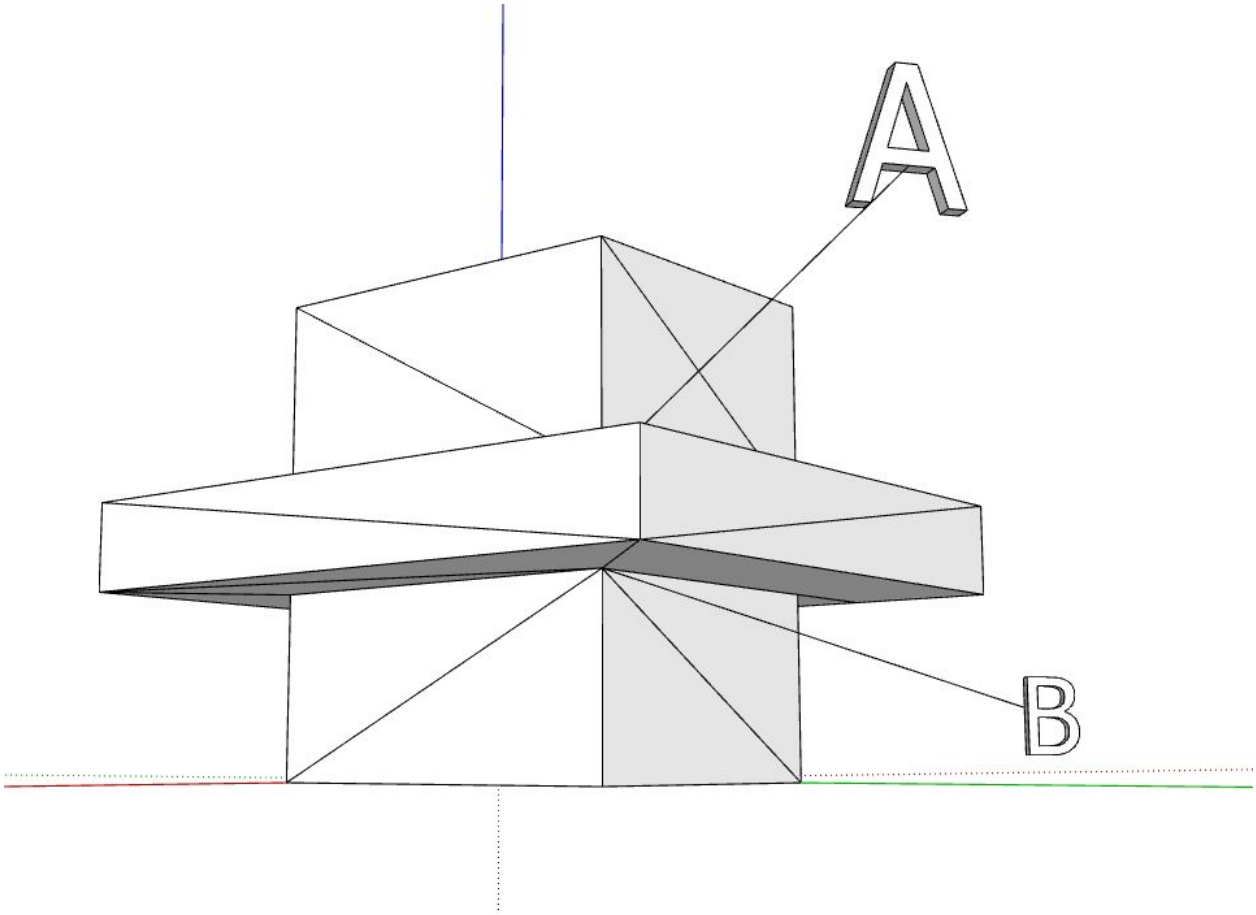
Where (Xi , Yi, Zi) are the new coordinates, (Xm, Ym, Zm) are the model coordinates and (Xa, Ya, Za) are the real position of the antenna in the spaceship.

List of Vertex

To make sure the drone will cover all the satellite, we just make a list of all the vertex in the model and tell KVN to go to each one.

List of unions

Look at this example. If KVN wants to go directly from point A to point B, he will find this action impossible, because in any direction he wants to go, the distance between he and point B will increase



The proposal solution for this problem is make KVN moves between edges, so to go from A to B will have to first go to C, then to D and then to A

Best way

To know which combination of points are the best to go from one point to another, the software ask to the first point , which others points are united to them, and if the objective point is it not attached to that point, randomly choose one point and do the same action, until the objective point is reached, when its reached its saves the path and count how many points have that way. The software make this protocol lots of times and then choose the one that has less points.

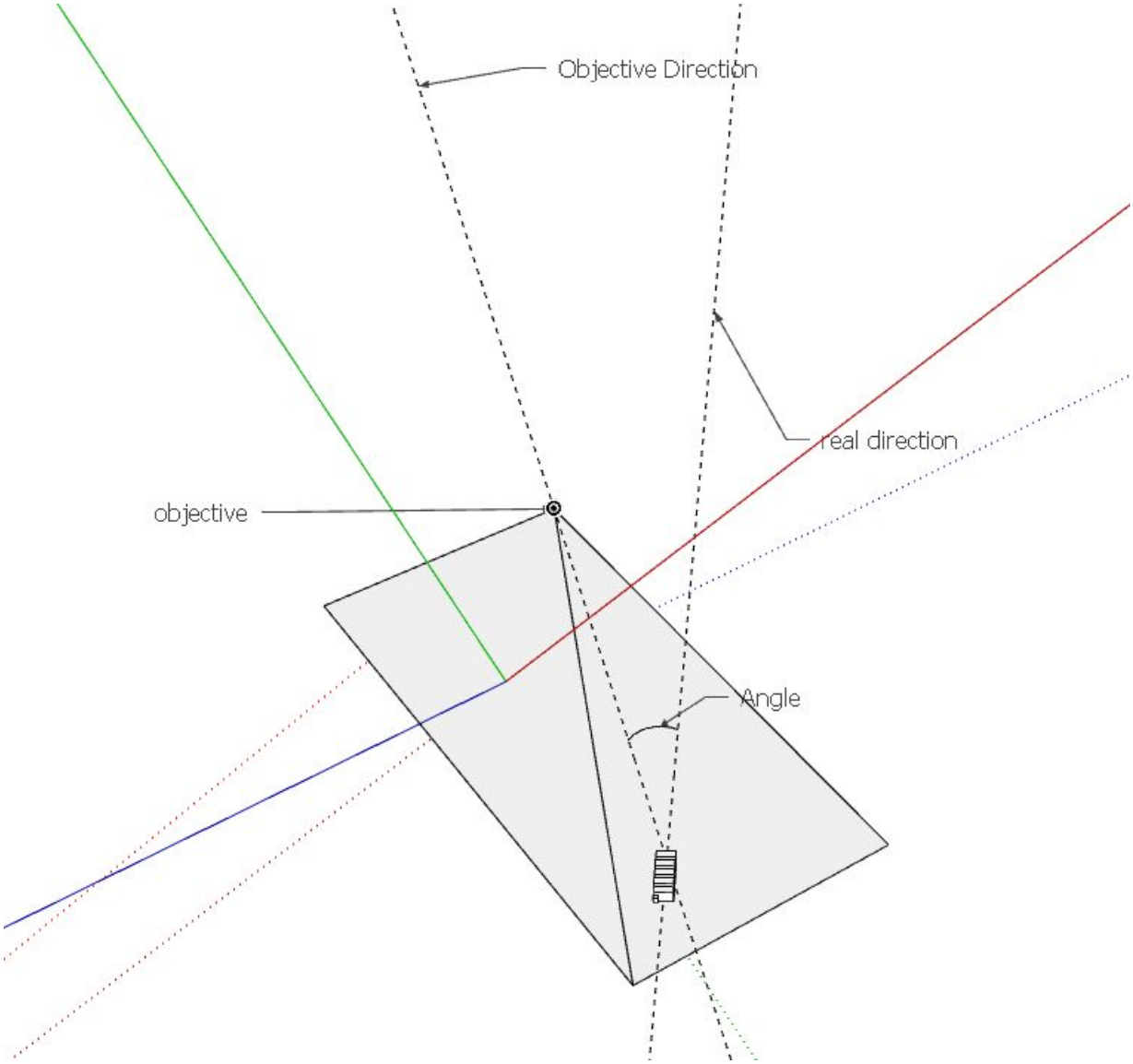
Direction

An important data for the drone to move is the angle between the schuss going from the dron to the objective and the schuss from the actual position of the dron to a previous position of the same dron (a second or less in the past)(schuss of the dron) ,to know which angle is rotated the dron we use this formula

$$cos(\alpha) = \frac{|u \cdot v|}{|u| \cdot |v|}$$

u and v are the vectors of the schuss

But this formula has a problem, it don´t tells if the angle is positive or negative so to correct this problem, we make an arbitrary schuss in the space and we calculate the angles between the objective schuss and the arbitrary schuss, and the angle between the dron schuss and the arbitrary one, then subtract both angles and we will know the angle and if it is positive or negative



Conclusion

To be able to progrma and test this software:

1. We need a local position hardware.
2. We need gecko grippers, to make our prototipe adhere to the surface.
3. Hopefully in a future we will have this technology and we will continue this project.