**Autonomous Object Coordinate Evaluation**

*Algorithm Description Document*

This document pertains to the evaluation of latitude-longitude coordinates of stationary objects on the earth from a sequence of images and a known set of parameters from the acquisition system.



///////////////////////////////////// OUTLINE //////////////////////////////////////////////////////////////

* Summary
* Inputs/Outputs
* Algorithm overview
  + Flow chart (top level)
  + Image Processing
    - Image recognition (blob detection, cluster)
    - Exceedences, false alarms, etc..
  + Object tracking
    - The association/assignment problem
    - Auction algorithm (Mugry’s? Greedy?)
    - A linear model of predictions
  + Coordinate calculations (This whole section might fall into tracking, but we will have to discuss SW architecture at some point
    - Possibly two methods?
      * Triangulation
      * Size
    - Filtering and error calculations
* Unit testing
* Appendix

Some notes:

Each block in a diagram will have an associated input/output list to keep things in line.

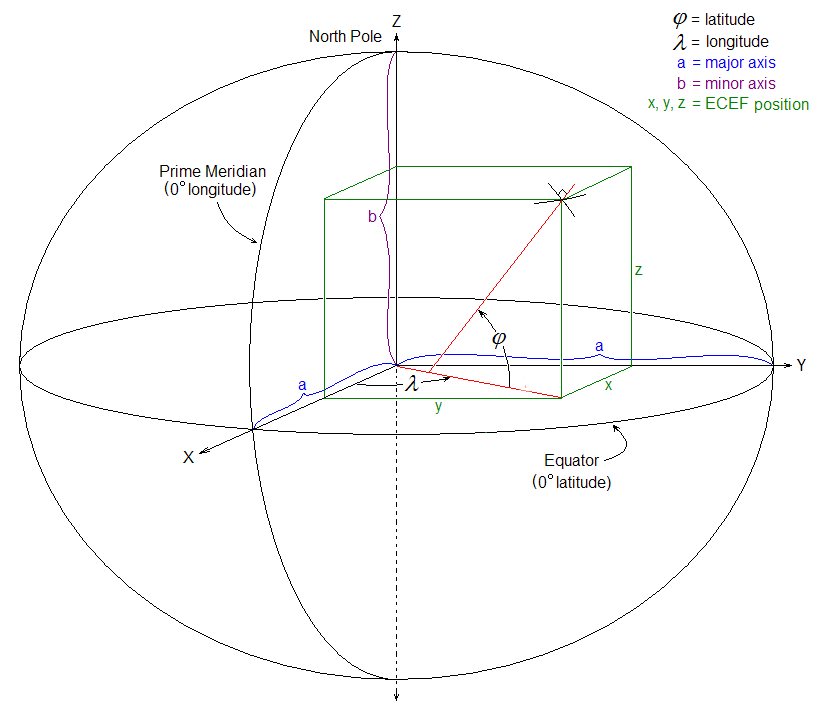
The analysis of each ‘block’ should be performed prior to thinking too much on the top down architecture.

We should really consider an object oriented approach.

///////////////////////////////////////// Document //////////////////////////////////////////////////////////////////////////////

**1. Coordinate Systems**

This algorithm document will refer to two earth coordinate systems: the Geographic spherical coordinate system, and the Earth-Centered-Earth-Fixed (ECEF) cartesian coordinate system. The Geographic system is the traditional geodetic latitude/longitude/elevation system. Latitude and longitude are both in degrees and elevation is in kilometers. The point where (lat,lon) = (0,0) is the intersection of the prime meridian and the equatorial plane. The ECEF system is an orthogonal 3D system. The X-axis is the line connecting the center of the earth with (lat,lon) = (0,0), the Z-axis is the line connecting the center of the earth with the north pole and the Y-axis is orthogonal to both X and Z and complete the right-hand coordinate system. Below is a representation of the Geographic system and the ECEF system together.



It is important to differentiate between the two coordinate systems. While inputs to the algorithm block will typically be in the Geographic system, the math will be cleaner and usually executed in the ECEF system, as the later descriptions will show. A simple set of transformations can be used to go from one coordinate system to the other and are provided with the WGS84 model.

Talk about this more in detail, maybe discuss the fact that it doesn’t matter which coordinate system we go to as long as we go back appropriately, although we will choose the geodetic to standard ECEF transformations since geodetic is provided by google and going to standard ECEF helps us a) validate and b) possibly use them for comparison with something else later.

**2. Algorithm Flow**

Insert a flowchart for the whole algorithm. Discuss timing which will be a critical issue. For example: image processing will execute for every image, while tracking and triangulation are only execute when something needs to be calculated (i.e. a new observation is passed from processing, thus they are event driven).

**3. Image Processing**

**4. Tracking**

“Tracking” is the interframe technique of associating a set of observations to a set of previously existing tracks. There are four main components of the tracking process flow and they are as follows:

* State Prediction
* Gating & the Cost Matrix
* Association
* Track Update

Process flow diagram here

**4.1 State Prediction**

State prediction is the process of predicting an estimate state of an object at time t+1 with knowledge of the object’s state up to, and at, time t. The state prediction, will be referred to from this point on as the location prediction since other states can be predicted and therefore utilized (i.e. object size, shape, color, etc…). Location has been chosen to maintain generality in this algorithm (e.g. not restricted to certain objects) although it is important to note that other states can be utilized at a later point if needed.

Note that we will do 2D derivations first, and make a note at the end about generalized to 3D with the normal vector and projections.

**4.1.1 Location model**

Location, in this model, will refer to a location on the sensor for a given frame in pixel coordinates (i.e. row, column), where the origin is the center of the upper left pixel.

Diagram

**Calculating the direction vector**

2D diagram of direction vector, LOS, and FOV

**Modeling of an object’s movement**

2D diagram of direction vector, LOS, and FOV and object (assuming WLOG LOS=direction vector)

2 sensor diagrams, time t and time t+1

**4.2 Gating and the Cost Matrix**

**4.3 Association**

**4.4 Track Update**

Below, are descriptions of the parameters and variables associated with the Tracking & Triangulation Interface

**Tracking Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Parameter Type** | **Parameter Size** | **Def. Frequency** |
|  |  |  |  |

**Tracking: Input Parameters**

**Tracking Input Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Name** | **Input Type** | **Input Size** | **Input Frequency** |
| Observation List | Double array | N x 3  (N: # of objects found in frame) | 1/frame |
|  |  |  |  |

**Tracking: Input Variables**

Observation List: Cast as a double array, this variable stores [row, column,

**Tracking Output Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Output Name** | **Output Type** | **Output Size** | **Output Frequency** |
|  |  |  |  |

**Tracking: Output Variables**

**3.1 The Association Problem**

**3.1.1 The Auction Algorithm**

**3.1.2 Modified Auction Algorithm**

This section aims to describe the required modifications of the chosen Auction to account for the frequent beginning of new tracks and end of old tracks.

**4. Triangulation**

**Triangulation Input Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Parameter Type** | **Parameter Size** | **Def. Frequency** |
|  |  |  |  |

**Triangulation: Input Parameters**

**Triangulation Input Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Name** | **Input Type** | **Input Size** | **Input Frequency** |
| Camera Coords | double | N x 2  (N: # of objects found in frame) | 1/frame |

**Triangulation: Input Variables**

**Triangulation Output Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Output Name** | **Output Type** | **Output Size** | **Output Frequency** |
|  |  |  |  |

**Tracking: Output Variables**