# Triangulation Algorithm

The triangulation algorithm allows the quad copter to determine its position over a landing pad with IR LEDs imbedded. The algorithm takes a set of two X,Y positions as inputs and outputs the quad copters x,y offset and elevation. The algorithm runs on the APM board in a 100Hz loop.

Algorithm Constants

The algorithm’s constants are determined from data gather from the Wii Camera Data sheet found at: <http://wiibrew.org/wiki/Wiimote>. These constants are used in many calculations and are camera specific. They are detailed below.

#define X\_PIX 1024 // number of pixels in X dimension of camera

#define Y\_PIX 768 // number of pixels in Y dimension of camera

#define X\_CENTRE X\_PIX/2 // X coordinate of Centre Pixel in Camera

#define Y\_CENTRE Y\_PIX/2 // Y coordinate of Centre Pixel in Camera

#define X\_FOV 47 // Field of view of Camera across X dimension

#define Y\_FOV 35 // Field of view of Camera across Y dimension #define PIX2DEG 0.045 //(X\_FOV / X\_PIX) # of pixels per deg of view

#define TARGET\_MAX\_WIDTH 200 //width of IR blobs on IR target in mm

Algorithm Inputs

The algorithm takes two inputs. These two inputs are Blob structs as defined in the camera code (see Library in Appendix XXXXXXXX). The structs contain the x and y pixel position of each blob being tracked.

// Structure to hold blob data

struct Blobs

{

int X;

int Y;

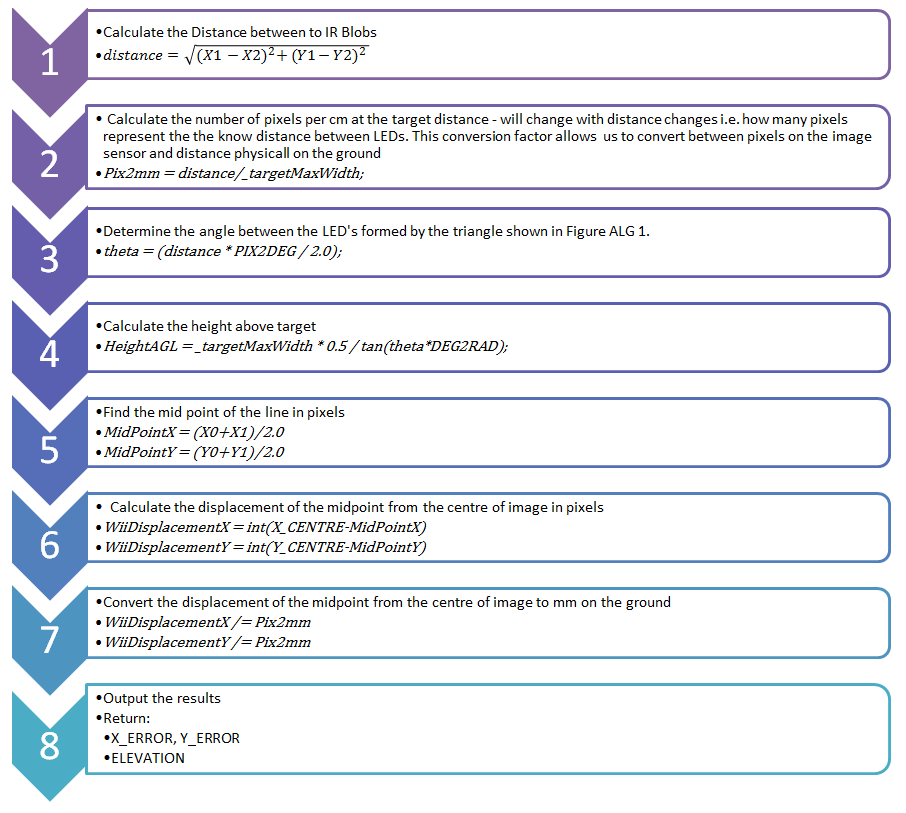
byte number;

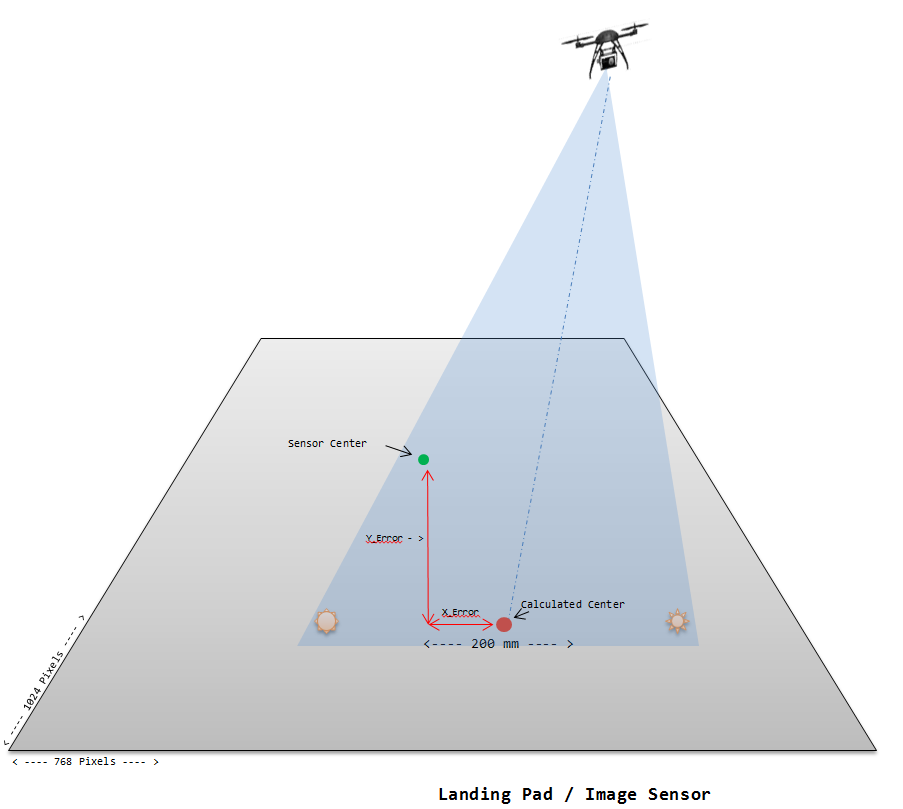
};

Algorithm Outputs

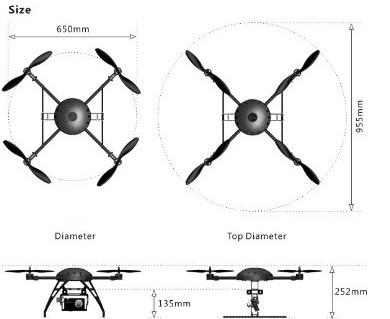
The algorithm output is the calculated location of the quad copter above the two LED blobs. This output is in terms of X and Y offset from the center of the blobs. These values are fed into the APM PID controller as X and Y position error. The PID uses these values to correct its position accordingly.

Algorithm Operational Flow- image version is below





Algorithm Geometry – word version, image version is above



<---- 200 mm ---- >

**Landing Pad / Image Sensor**

< ---- 768 Pixels ---- >

< ---- 1024 Pixels ---- >

Sensor Center

X\_Error

Y\_Error - >

Calculated Center