Autonomous Remote Control Drone

Author: Gary Kean

Email: gkean13@gmail.com

5/18/2014

Summary

There are many aspects to Remote Controls Drones that are not described well in any of the literature. Even after reading through much of the literature and relying on much of my personal knowledge and experience with Remote Control Airplanes, there was much about the mechanics, technologies, and outputs related to Drones that I didn’t understand. This paper documents my process of learning more about the mechanics and technologies of flying Remote Control Drones. I follows the preparation of the airplane for flight, the trials and tribulations of technology calibration, captured images and their qualities, and the process of processing the images for the final image rectification. The paper concludes with a description of the image rectification, lessons learned, and potential future efforts.

Purpose

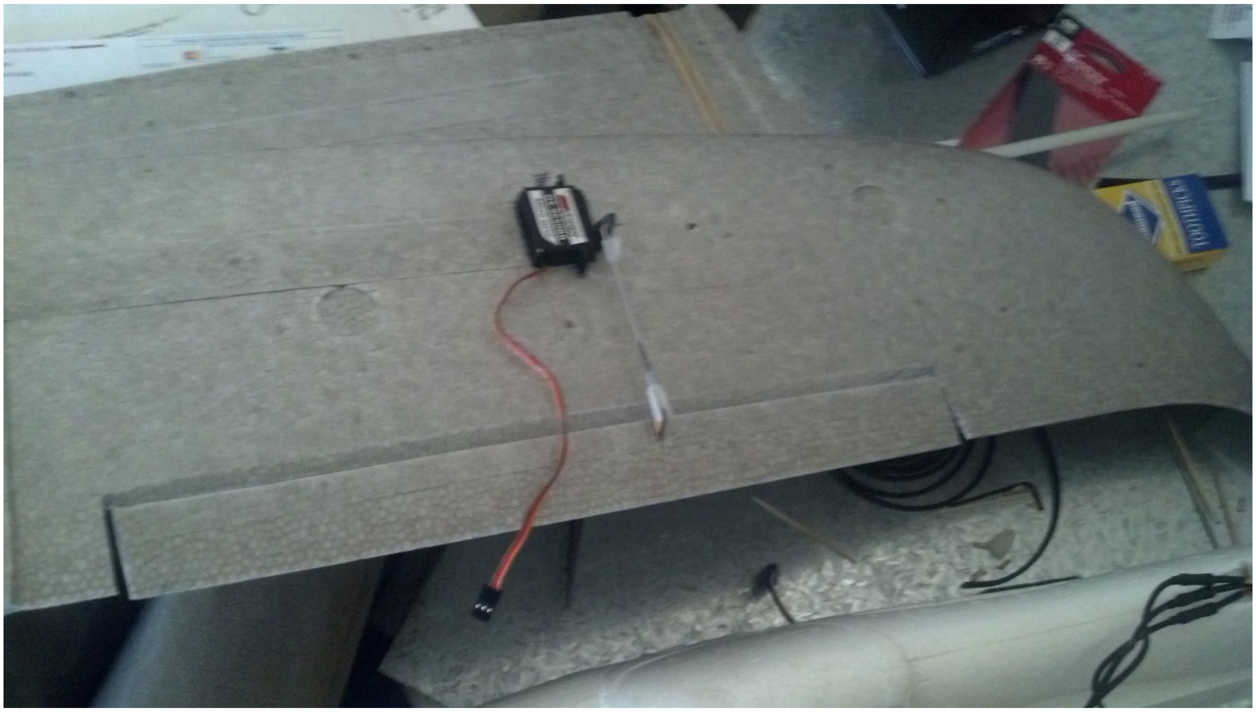
- ❖ Learn the technologies required for autonomous flight
- ❖ Capture and process images to be used and rectified to a Base map of the area

Equipment and Methods

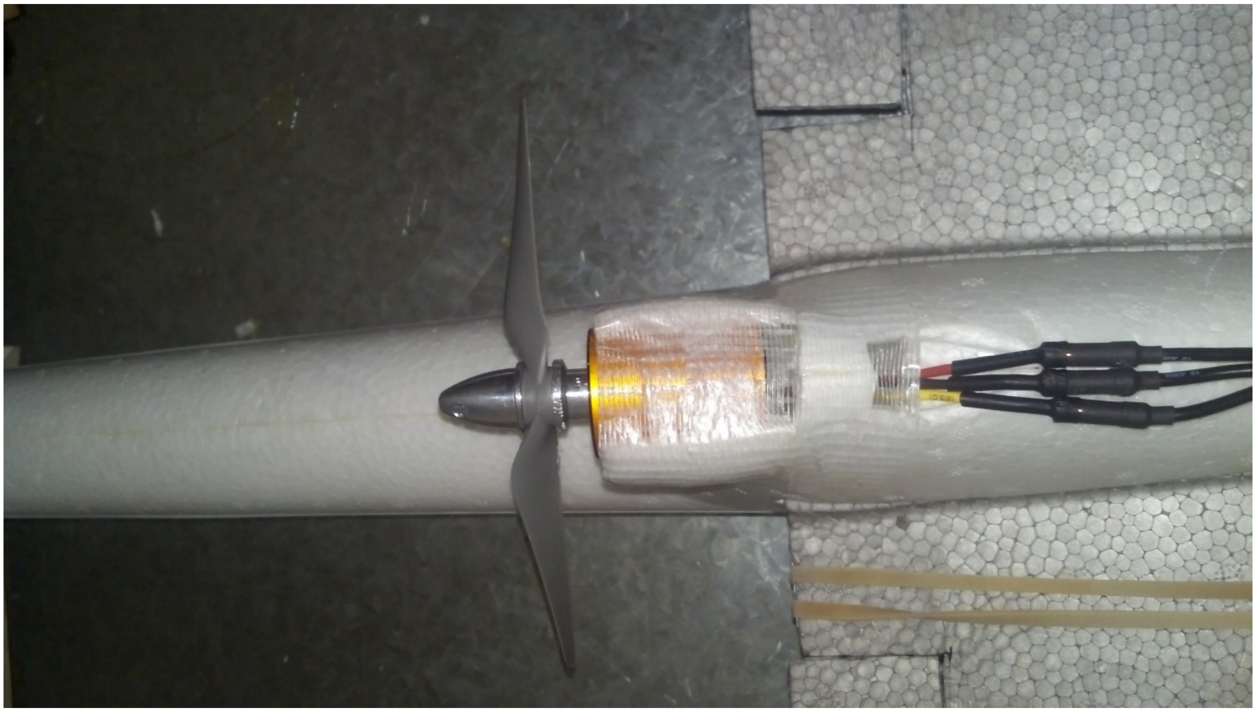
The Remote Control (henceforward RC) airplane used for this project was the Multiplex EasyStar. It is a rugged, very stable flying plane. It is purchased with the minimum of 3 RC controls – rudder to steer right and left, elevator to make the plane go up and down, and throttle control to control its speed.

Multiplex EasyStar

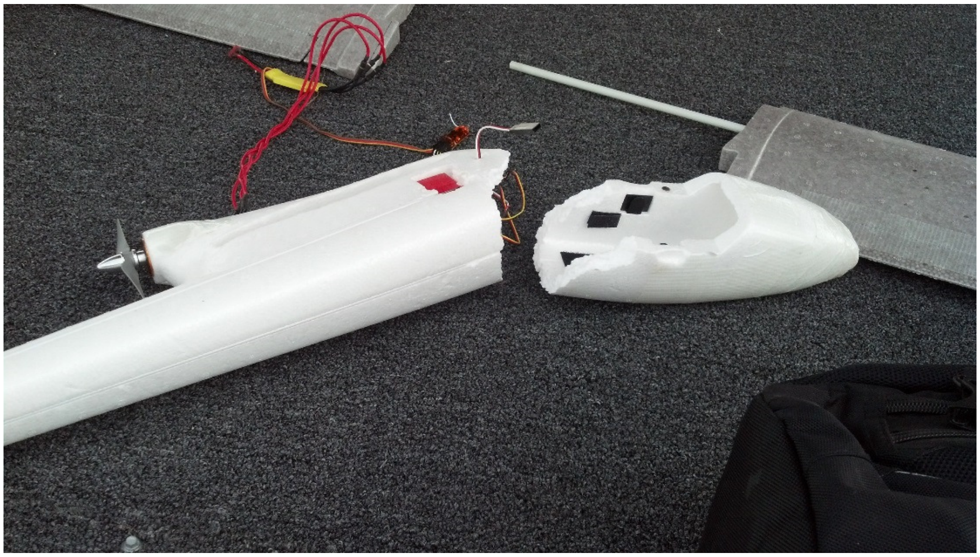
For better steering control, ailerons were added to the wings. This allows for sharper turning than is possible with just a rudder.



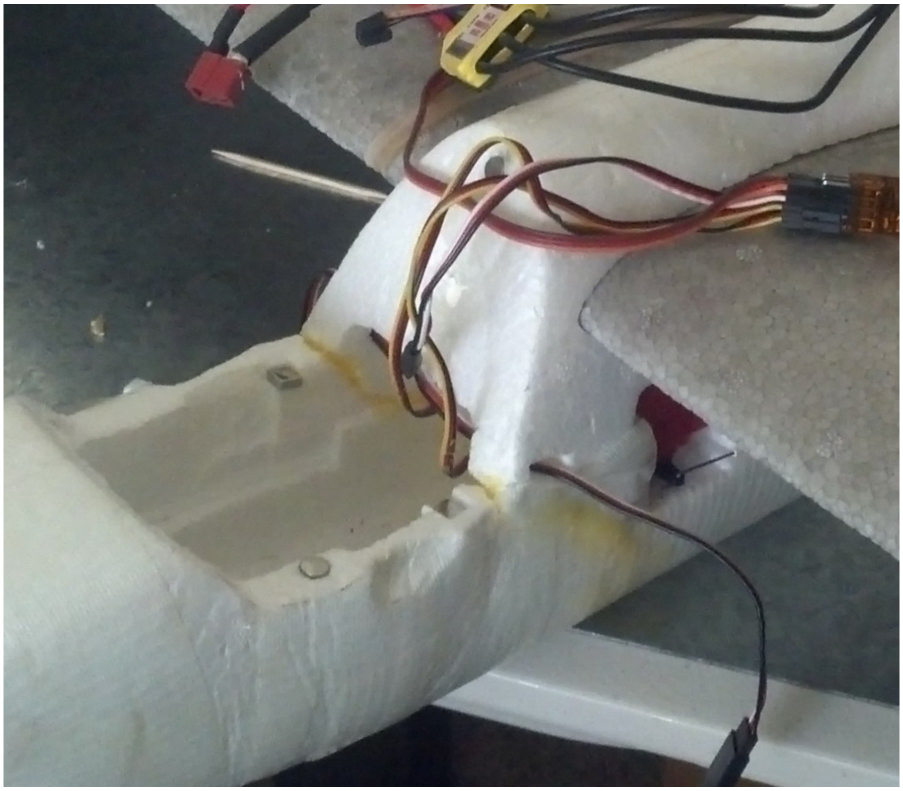
In addition to the ailerons, an upgraded motor and electronic speed controller were installed. The new motor is to give the plane additional power to compensate for the additional weight of the aileron servos, camera, and autonomous flight electronics.



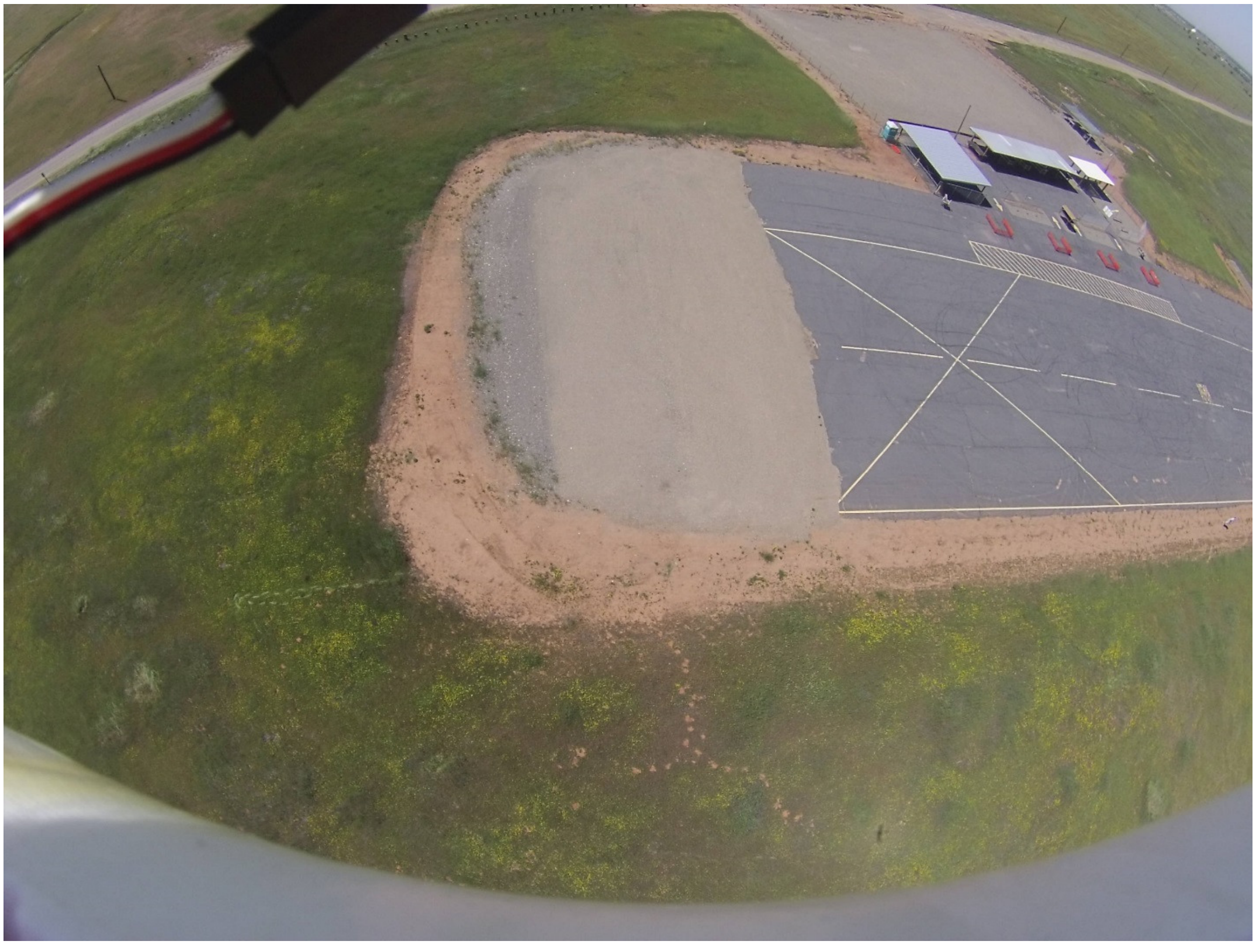
At this point, the EasyStar appears to ready for its first flight. I take it to an RC airbase in Rancho Cordova, CA. Batteries are charged and the controls look like they are working well. The weather looks good so I decide to take it up for a test flight and see how it handles. Unfortunately, what I thought were properly working controls were not!



A very, very short flight later, this is the result. It turned out that the control for the elevator was reversed and up was down and down was up and it was all over before I realized it. So, back to the bench to see if it can be fixed.



It looks like the glue worked and the elevator controls corrected, we are ready for another try. I have set the GoPro camera to take one picture every 10 seconds.

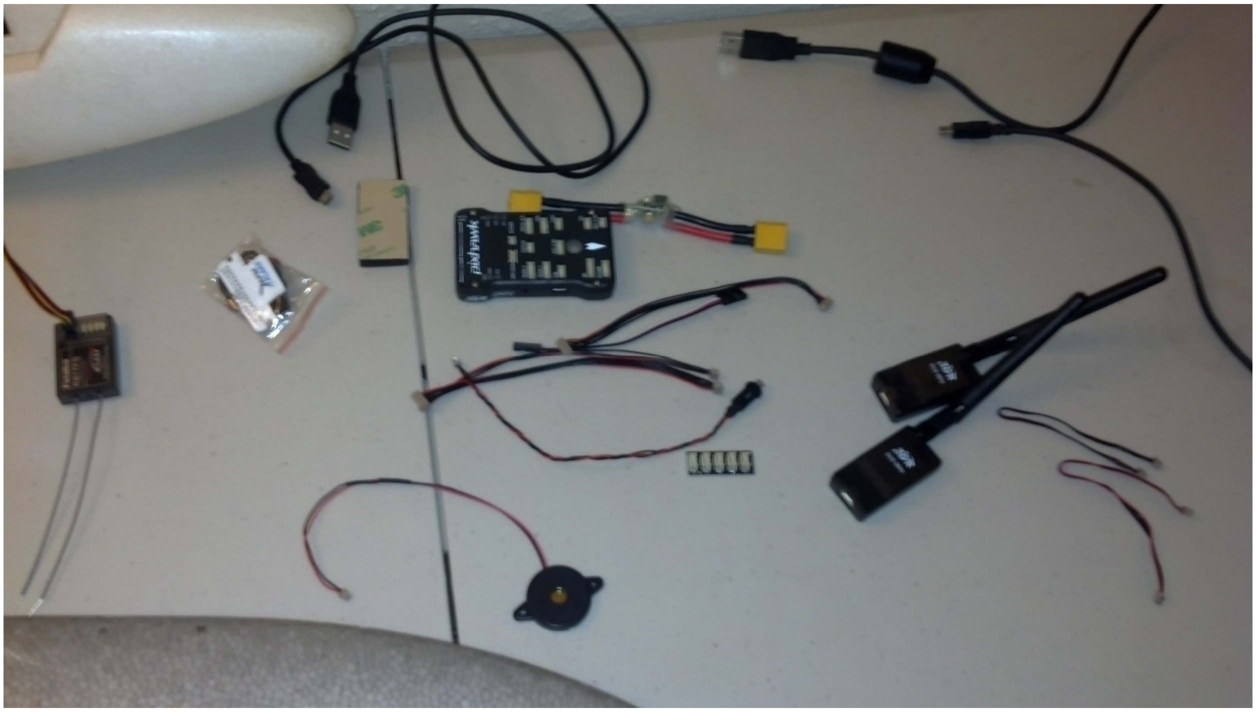


My first manual test flight to see how the plane handles.

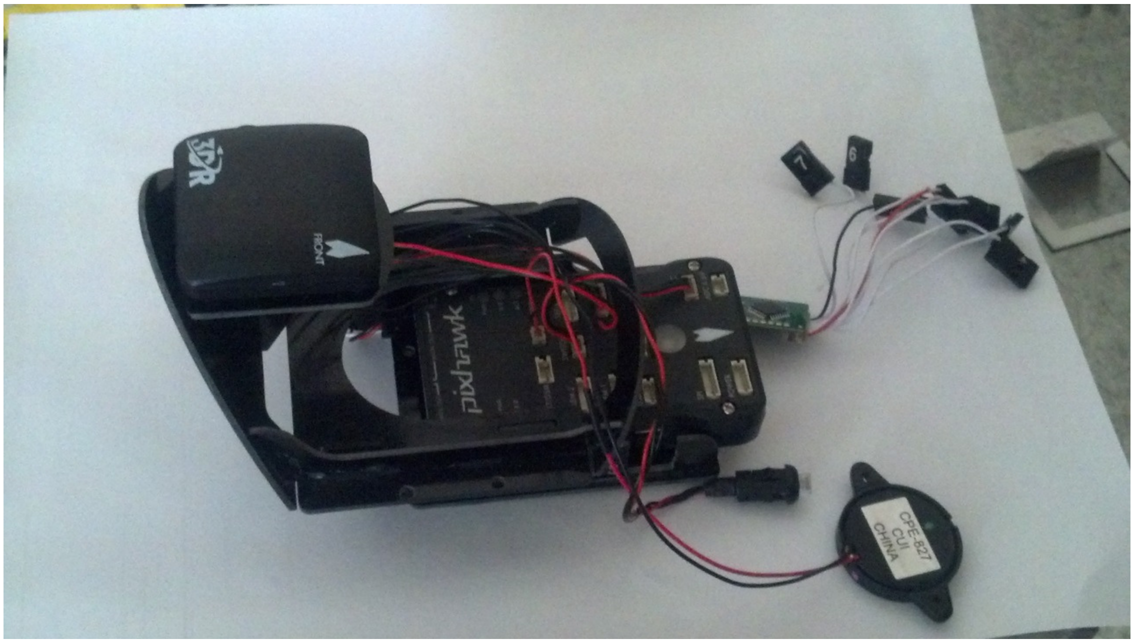
The planes is handling well, but I realize the camera is taking pictures too infrequently. For the next flight, I decide I want to set the camera to take a picture every 5 seconds because I feel like the speed of the plane is such that I am not getting enough overlap of the imagery to able to produce a good mosaic.

Autonomous Flight Hardware

Now that I have a plane that is flying well, it is time to prepare the autonomous hardware. The system I have chosen to use for this project is the PixHawk from 3D Robotics. It is small and lightweight and built specifically to work with the Multiplex EasyStar or the HobbyKing Bixler. There are quite a few components that ship with the PixHawk.



When all assembled correctly and mounted on the mounting bracket, it doesn't look quite to daunting.



Top left – GPS unit

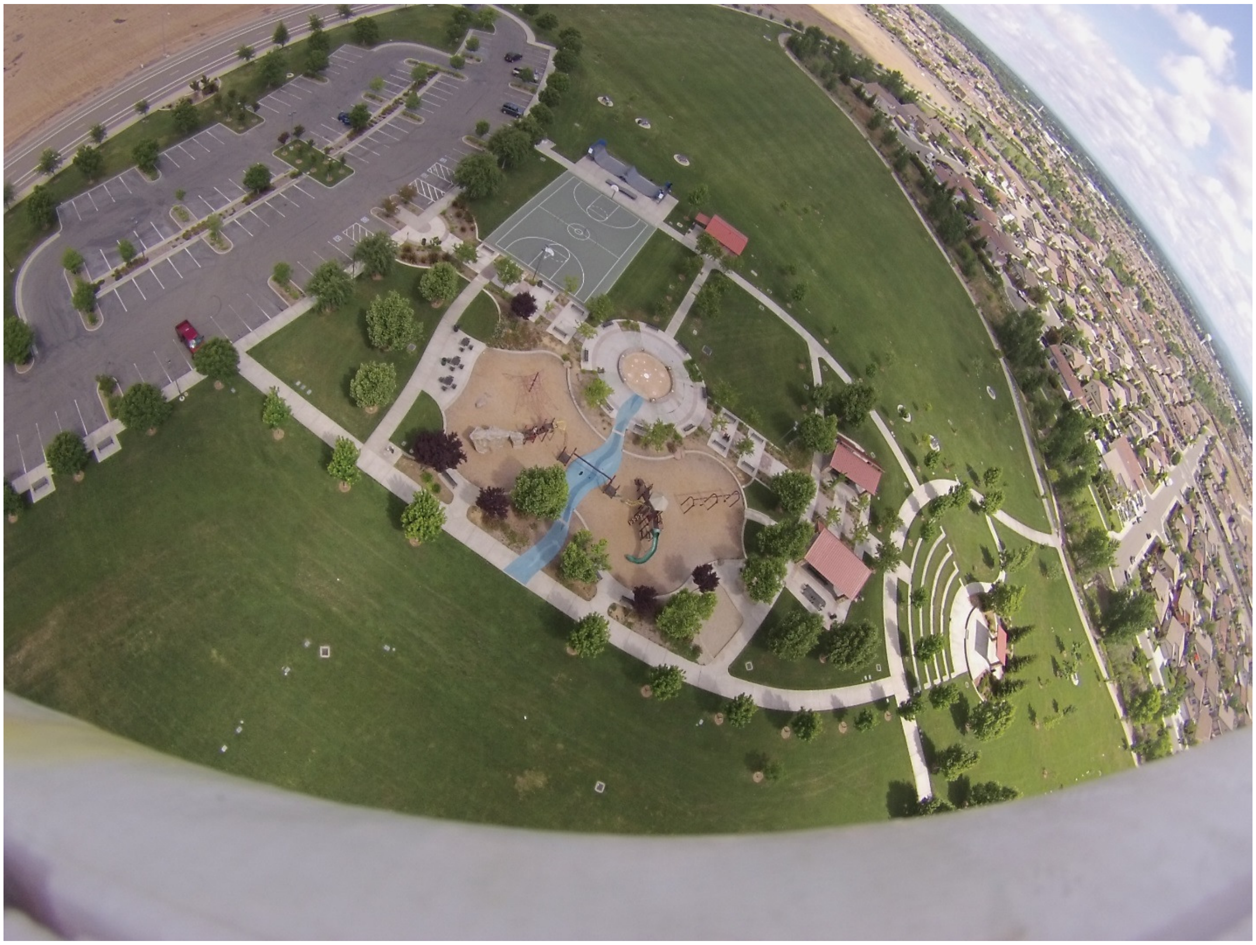
Middle of image – PixHawk device

Top right – PPM Encoder that connects to RC receiver

Bottom right – Buzzer – for tracking if plane is downed out of your sight

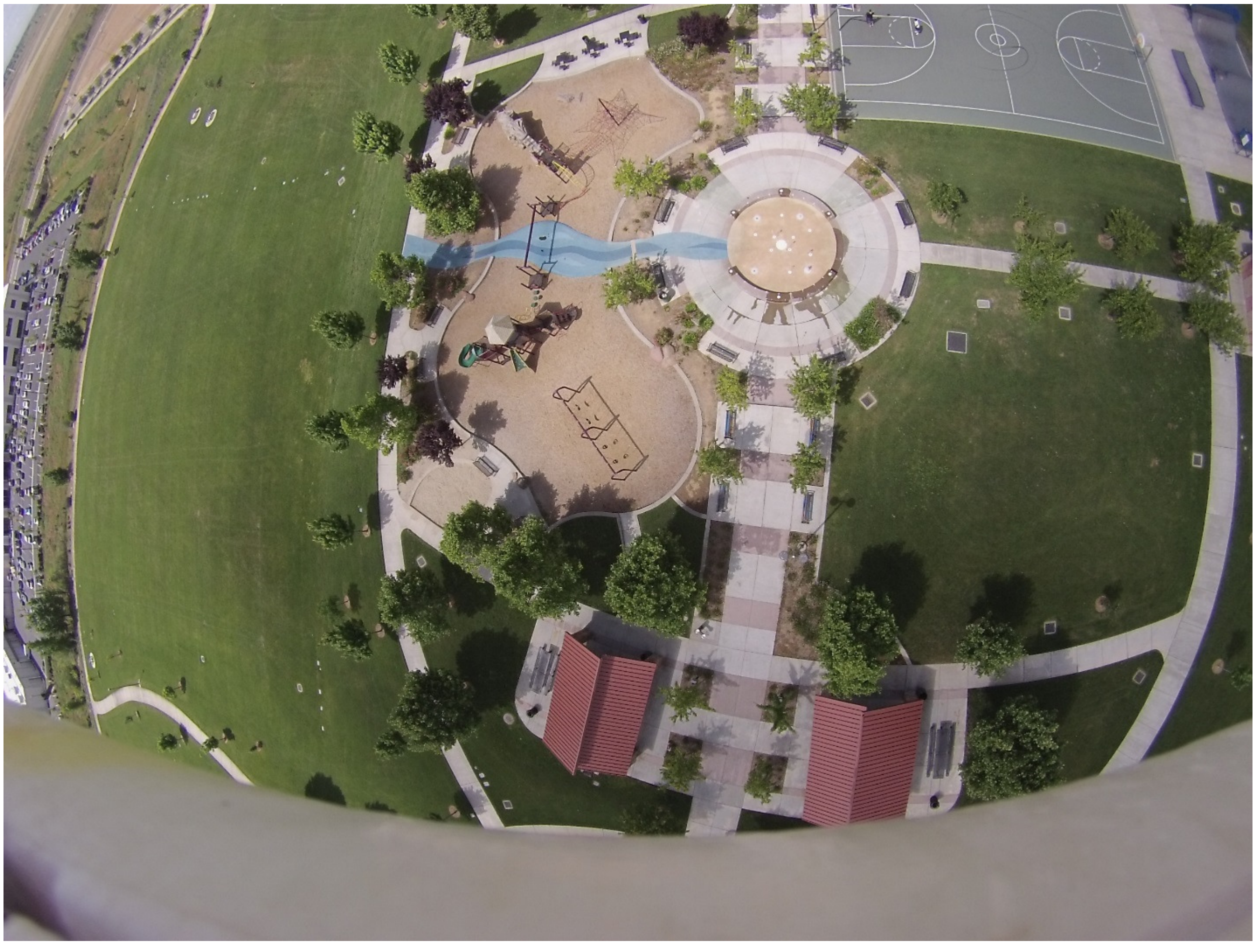
To see if the system is working correctly, I run it through the step-by-step calibration process before I install it in the airplane. The process has the user move the system in a 360 degree sphere of motion so the software can understand and calibrate what each position looks like to it. Once the calibration was completed, outside the plane, I moved on to installing the system in the plane and running the calibration again. Unfortunately, when it was all installed in the plane, I could no longer get the software to calibrate the hardware correctly. I tried many times and even called tech support at 3D Robotics to see if they had any ideas about what could be going wrong. I was not able to get the calibration to work correctly again and was running out of time to complete the project. I made the decision to do the flying of the play manually so that I could get some images the rectification portion of my project.

Stone Creek Park, Rancho Cordova, CA



After having taken quite a few images at the RC park in Rancho Cordova, I realized that there was a great deal of grassland with almost no reference points to make a mosaic or to use in the rectification process. Because of this, I chose to fly the plane at the Stone Creek Park in Rancho Cordova. This park has many trees, distinctive play area and driveway shapes, and it butts up behind a subdivision which also some interesting shapes to match against.

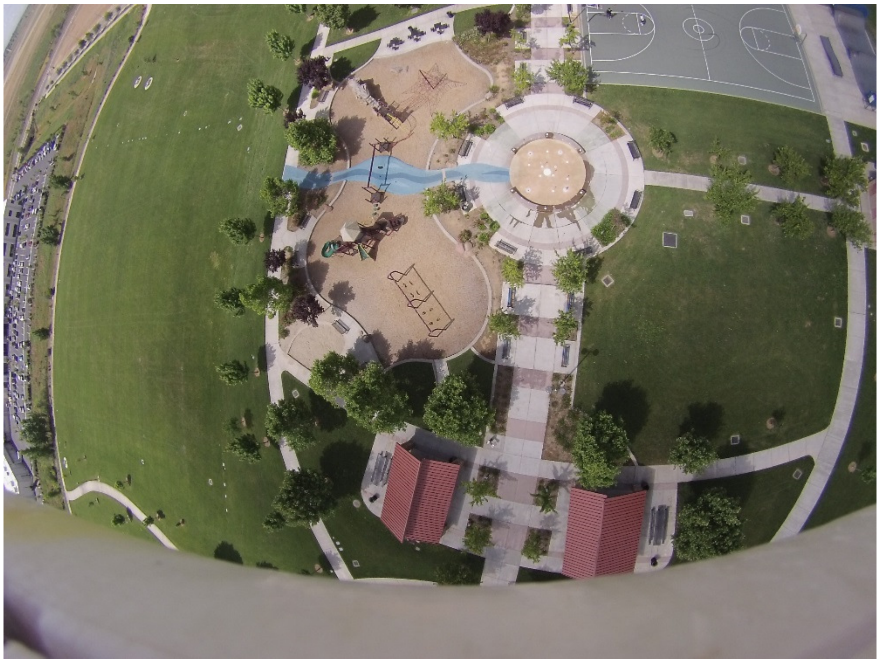
At the park, I made two flights, for a total of about 32 minutes. During that time, with the camera set to take a picture every 5 seconds, I captured almost 375 images. I realized when I got the images home and started looking through them, that there were a few problems with the images. First of all, because I had to fly the plane manually, it was very difficult to keep the plane flying straight and level for very long. So what I found was that only between 20 and 30 of the images were taken at an appropriate straight and level attitude. The second problem came when I started looking at the images more closely and realized that the camera comes with a fairly severe fish-eye lens that caused major distortion.



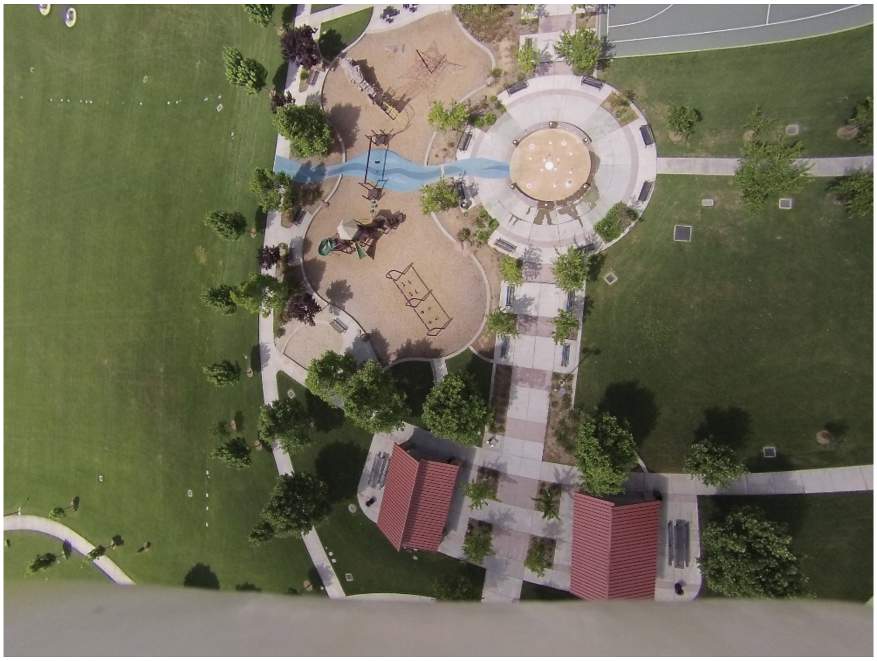
Although this image is taken from a fairly level attitude, there is nothing realistic about the distortion created by the camera lens. I realized that I was going to have to find a program that would be able to flatten these images before I could use them for any sort of rectification.

GIMP – Image manipulation software

I found a valuable asset in the Image manipulation software called GIMPSHOP 2.8. It is a free, open source application built to compete with Adobe Photoshop. GIMP has a built-in filter called 'Lens Distortion.'

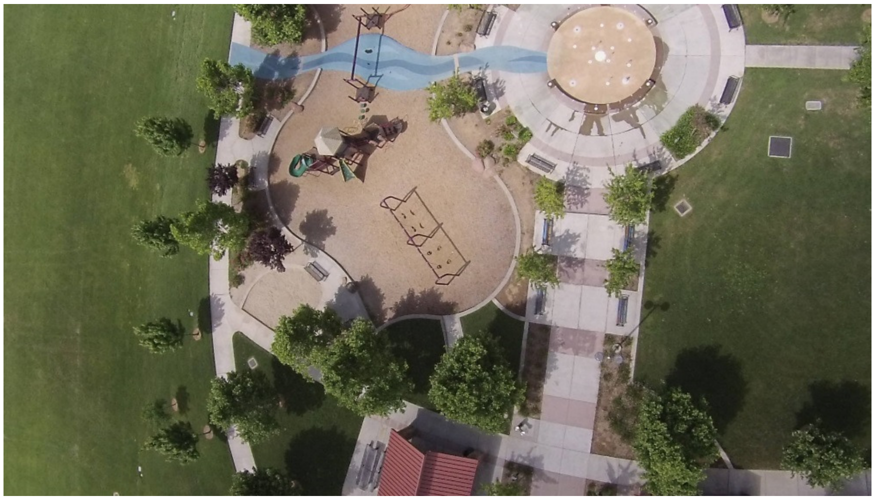


Raw Image



Same image with 'Lens Distortion' applied

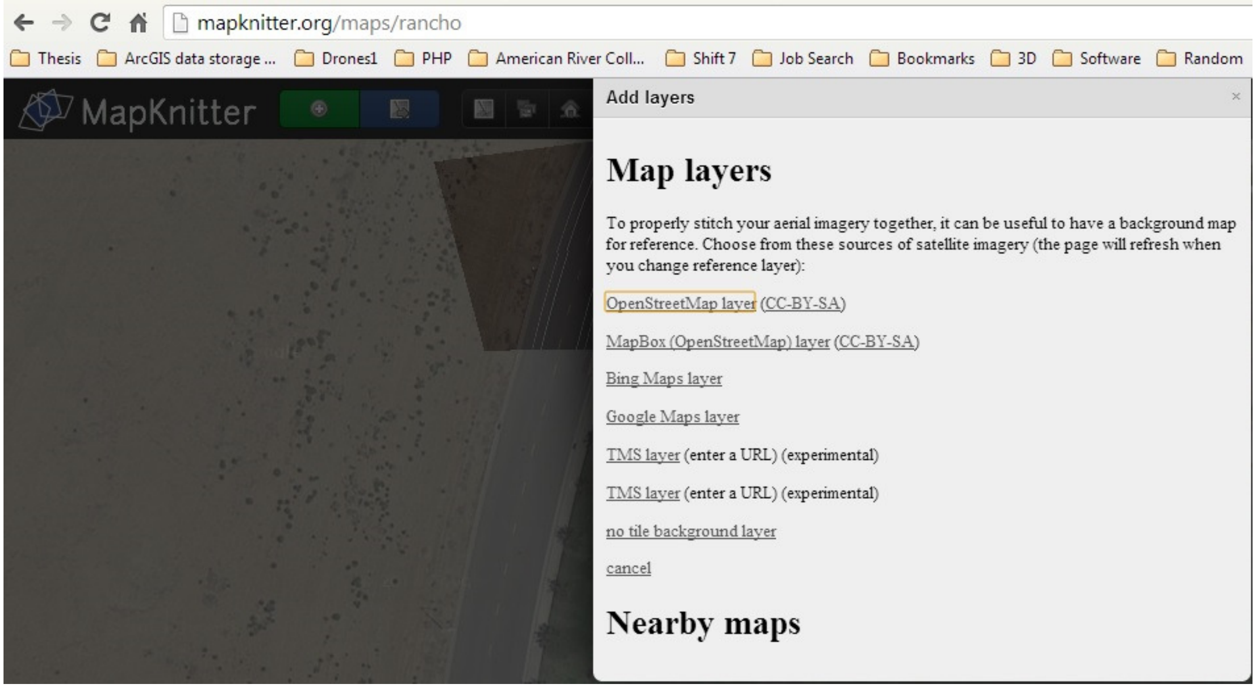
Unfortunately, as you can see in the images above, although the image is flatten out fairly well, it is also clipped around the edges and you lose about 25-30% of the image to ‘fall-off’. Not only that, but as I was working on the rectification process, I found that as well as the filter works, it works best on the middle portion of the image, while the edges are left slightly skewed.



This required me to clip out the middle 30-40% of the images for my final rectification process.

MapKnitter.org

MapKnitter.org is a web page that allows a user to import aerial photographs and then manipulate their shape, size and scale in order to overlay them on a basemap. MapKnitter has several options for choices of basemaps.



Add layers

×

Map layers

To properly stitch your aerial imagery together, it can be useful to have a background map for reference. Choose from these sources of satellite imagery (the page will refresh when you change reference layer):

OpenStreetMap layer (CC-BY-SA)

MapBox (OpenStreetMap) layer (CC-BY-SA)

Bing Maps layer

Google Maps layer

TMS layer (enter a URL) (experimental)

TMS layer (enter a URL) (experimental)

no tile background layer

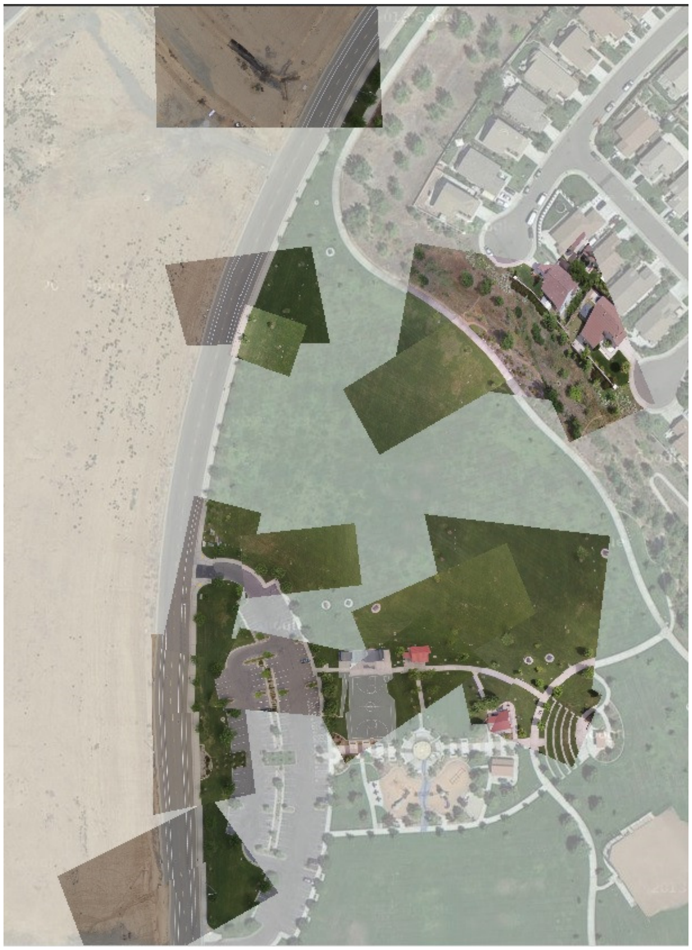
cancel

Nearby maps

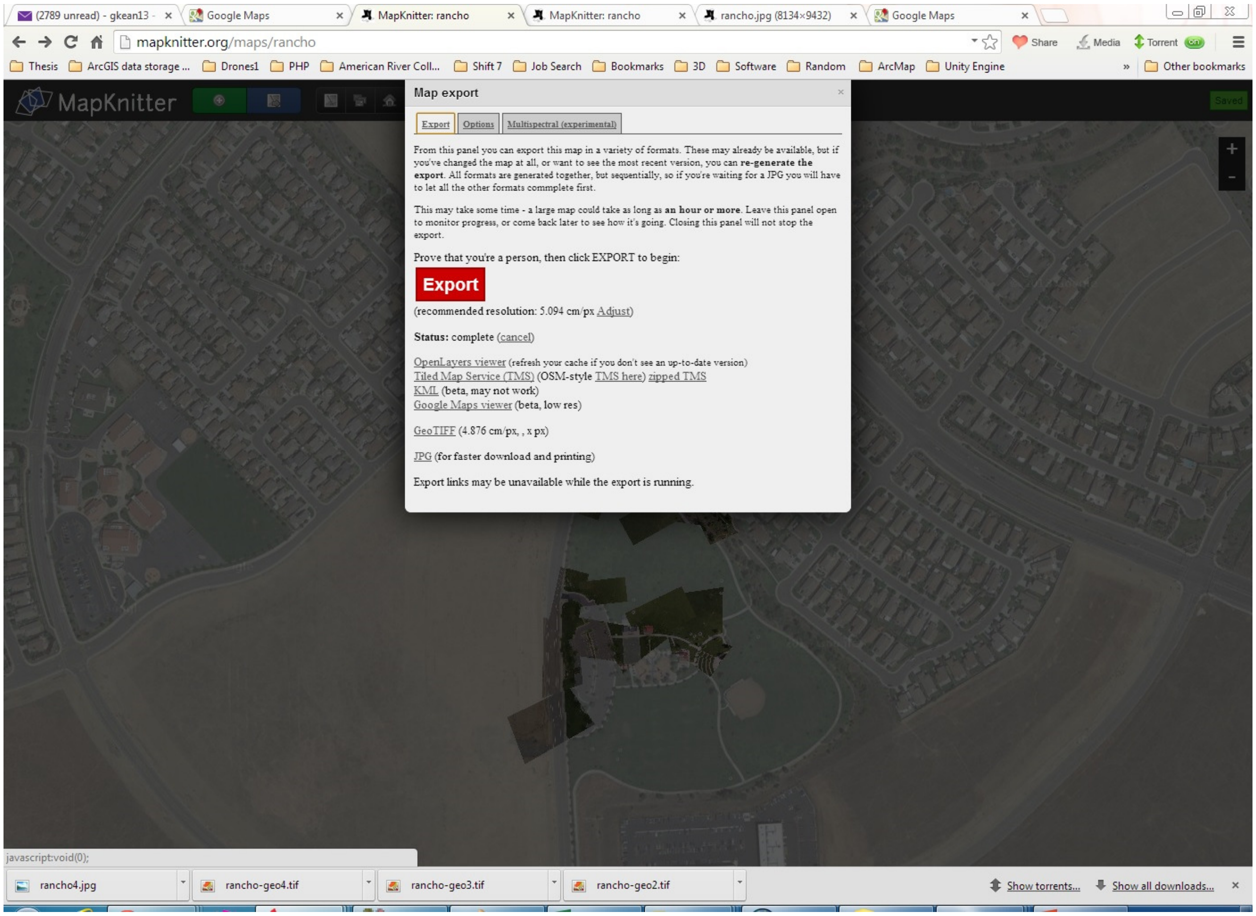
For my project I chose to use a Google map layer. I felt that the Google map was one of the better quality maps of this area.

Now that I had as many of the filtered images completed as I thought were valuable, I started importing them into MapKnitter. Each image needed to be moved, rotated, and scaled to get it close to its final position. MapKnitter has a nice feature in that it allows you to make your images semi-transparent to help with alignment. This feature really helped to dial in the location for each image. It also has a feature that allows the user to hide an image completely in the event that two images use the same alignment points. Although it is easy to use and understand, I had some issues with images shifting, requiring me to readjust all the images to their appropriate locations. The fix for this problems seemed to be to make sure the images weren't 'locked' in place. The 'lock' feature is intended to keep you from accidentally moving an image you have successfully placed. However, when zooming the map to get a better view of your reference points, somehow the 'locked' images all seem to shift slightly causing the need to readjust them again. It was frustrating until I figured out a solution.

My final image in MapKnitter consisted of 13 of the images I deemed useful. This is the final layout.



Finally, I had enough images placed that it was time to export them image and start the image rectification process. As with the basemap options, there were several options for exporting the final image.



Map export

Export | Options | Multispectral (experimental)

From this panel you can export this map in a variety of formats. These may already be available, but if you've changed the map at all, or want to see the most recent version, you can **re-generate the export**. All formats are generated together, but sequentially, so if you're waiting for a JPG you will have to let all the other formats commplete first.

This may take some time - a large map could take as long as **an hour or more**. Leave this panel open to monitor progress, or come back later to see how it's going. Closing this panel will not stop the export.

Prove that you're a person, then click EXPORT to begin:

Export

(recommended resolution: 5.094 cm/px Adjust)

Status: complete (cancel)

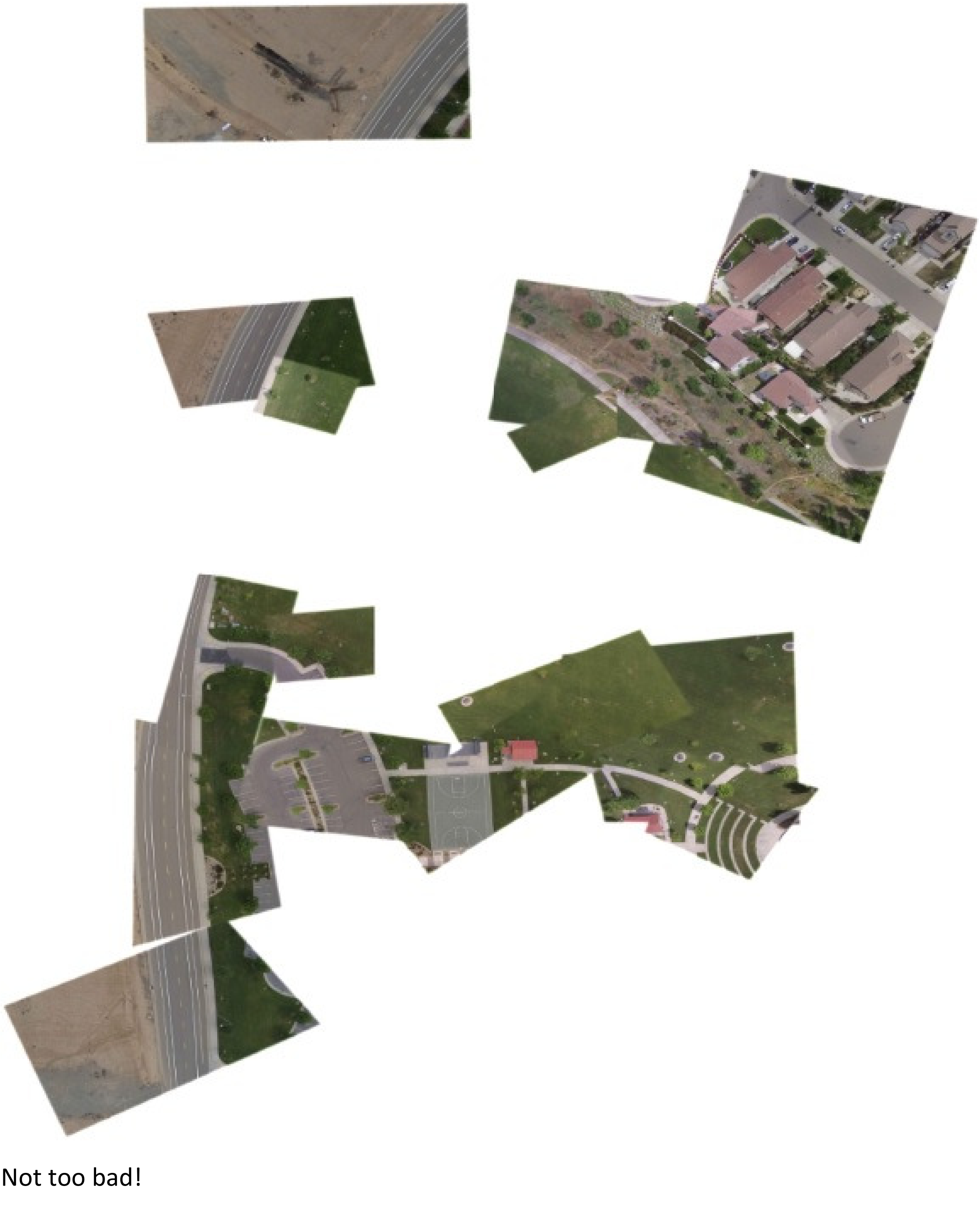
OpenLayers viewer (refresh your cache if you don't see an up-to-date version)
Tiled Map Service (TMS) (OSM-style TMS here) zipped TMS
KML (beta, may not work)
Google Maps viewer (beta, low res)

GeoTIFF (4.876 cm/px, , x px)

JPG (for faster download and printing)

Export links may be unavailable while the export is running.

The option that seemed to make the most sense was the GeoTIFF. I suspected I wanted to use a TIFF image in the rectification process and I thought that the GeoTIFF was MapKnitter's proprietary naming and formatting convention. I exported the image and this is how the final product looked.

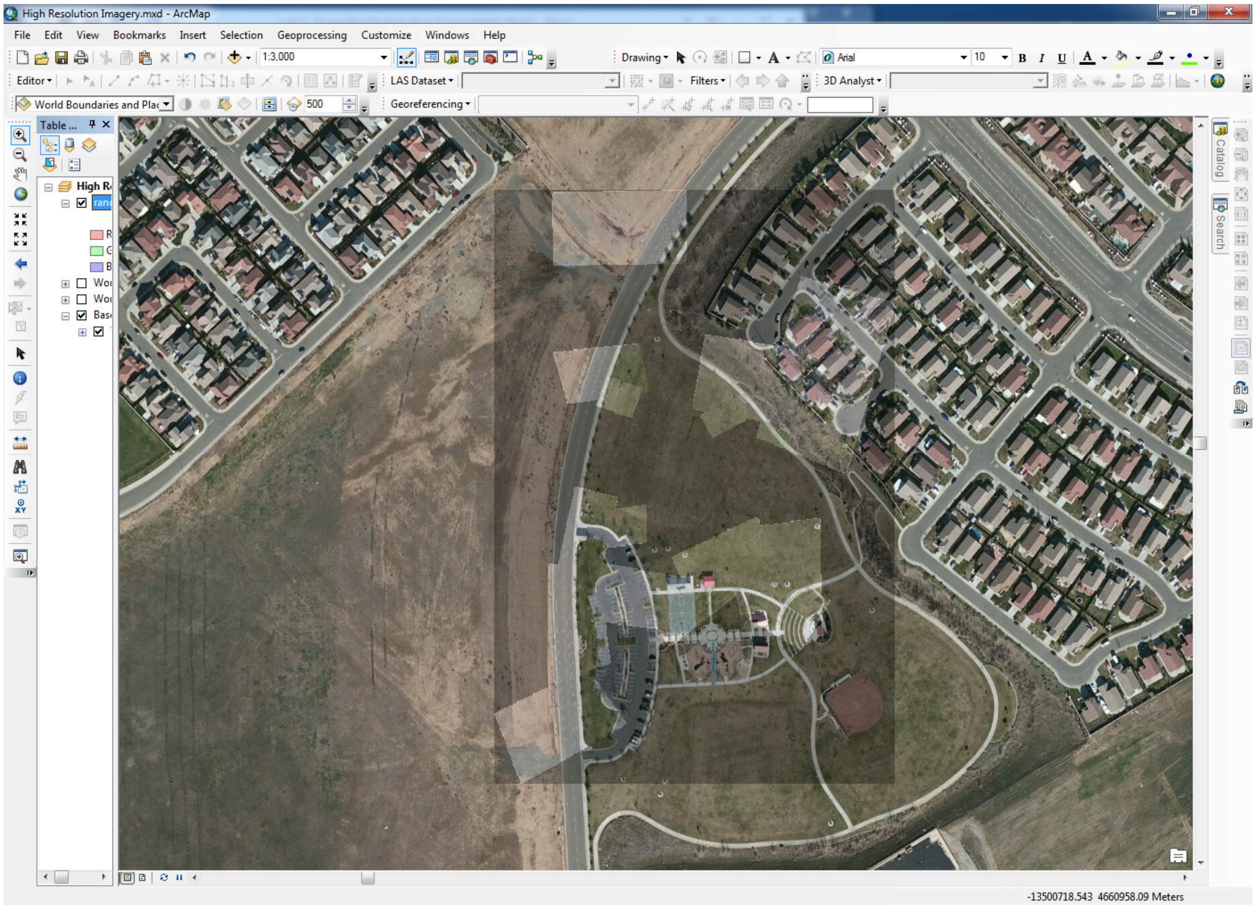


Not too bad!

Image Rectification – ESRI ArcMap 10.1

The final step to this whole process is the Image Rectification. My plan was to import the image into ArcMap and perform the process we went through in class. This required finding 5 reference points around the image to scale the image appropriately.

In ArcMap, I loaded a background reference image from ArcGIS Online of the Stone Creek Park area of Rancho Cordova. I zoomed in to the approximate area of my exported image from MapKnitter. I clicked on the 'Add Data' function in ArcMap and loaded my image of the park. I dropped into the image almost perfectly – correctly rectified!!!!


High Resolution Imagery.mxd - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:3,000
Drawing
Arial
10
Editor
LAS Dataset
Filters
3D Analyst
World Boundaries and Plac
500
Georeferencing
Table ...
High R
Catalog
Search
-13500718.543 4660958.09 Meters

Black background of the final MapKnitter file made visible with transparency to show the information being rectified.

Apparently the 'GeoTIFF' format from the MapKnitter application contains georeferencing information in the file that is read correctly by ArcMap, allowing for rectification process to occur automatically.

Conclusions

This was a very exciting project for because it allowed me to combine school with one of my other interests, RC airplanes. I was hoping that the software and hardware I used to make the airplane autonomous would work correctly. There are plenty of success stories to read about on the 3D Robotics website. I have not completely given up on trying to make the technology work. I hope, in the coming months to find the time to revisit the calibration process with some of the additional information I received from the 3d Robotics technical support and have a more successful attempt.

The MapKnitter website is a well-designed application with some very powerful abilities. Its' ability allow the user to rotate, scale and skew their images to easily create image mosaics is incredible. I'm not sure how I would have been able to successfully produce the mosaic I did with any other imaging program. The tools and ease of use are unmatched and I am very excited about the results I was able to produce. Their ability to imbed geographic data in the exported file that made the Image Rectification so simple for me was just so unexpected. I will definitely be keeping the URL in my back pocket for any future needs, I suspect it will only get better.

References

https://store.3drobotics.com/products/3dr-pixhawk

http://www.hobbyking.com/hobbyking/store/__16544__bixler_v1_1_epo_1400mm_arf_.html

http://www.gimpshop.com/gimp/?gclid=COzv14Ooub4CFc-BfgodLgoAIg

http://mapknitter.org/