Moving Object Tracking in Aerial Imagery

Sagar Arora & Aditya Gupta

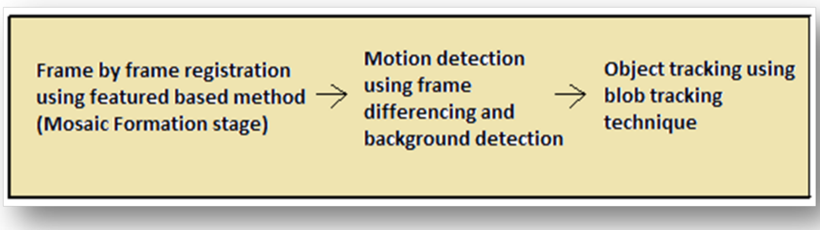
April 13, 2015

1 Objective and Aim of the Project:

Unmanned aerial vehicles are now day's very much popular in regard of their aerial photography and video (Taking images of ground from elevated position). Which is useful in many areas such as surveillance, Law and enforcement, spying, reconnaissance. For this rapidly growing technology and research field we have chosen a system called as COCOA developed by Saad Ali and Mubarak Shah (University of central Florida). This system is divided in several modules as:

- Platform motion compensation. (Ego motion Compensation)
- Motion detection
- Object detection

Main aim of this system is to perform multiple moving object tracking and path estimation (With the help of the mosaic technique). Block wise working of this system is explained in later stages. Block diagram of this system is: [This block is in reference with the system (COCOA)]



2 Image registration (Stage 1)

Image registration basically is the process of transforming different sets of data into one coordinate system. So ideally the telemetry data is registered with help of stored data (reference data) but as we didnt have telemetry data, we took 2 adjacent images of data set and considered one as telemetry image and other reference image. In the paper the author used two approached to do the same i) Feature Based ii) Gradient Based. It used both of them as using either of them taken one at a time had certain pros and cons so it fused the two methods. Feature based method is invariant to the strength of the geometric deformation, while it fails with weakly textured image content. While other method doesn't have that drawback but instead it need a good initialization when the deformation is strong. So when the first method is applied here we have used Harris corner Technique. To find the similar features of the images in the data set. This how we are registering different set of images. Mosaicing was done with combination of two methods feature based and gradient based. Actually using single one of them has their own drawbacks. If feature based used alone used it is invariant to the strength of the geometric deformation, while it may fail when the image content is weakly-textured, periodic etc. On the other hand, area-based (direct) alignment methods are able to align such images, while they need a good initialization when the deformation is strong, in order to avoid being trapped in local extrema. Therefore, combining the methods might be a better choice in some cases. So the toolbox provide us with the mosaic. In feature based method we compare two images to be compared and label the common features in both of them and later on these features are thresholding using RANSAC function.

Now the the first input for the mosaic was the first image in the data set then for the forthcoming frame reference frame was the mosaic image. So frames keeps on stitching and it forms a mosaic.

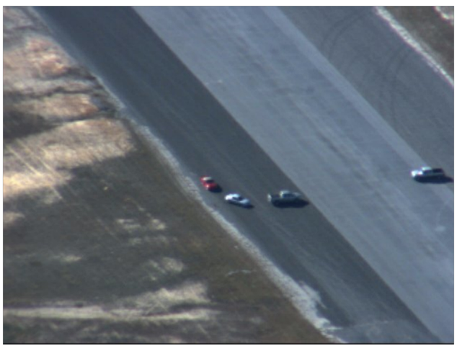


Figure 1: A sample image.

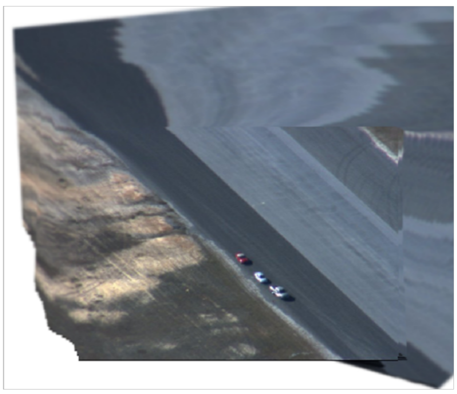


Figure 2: The final mosaic.

Functions

1. [INLIERS, MODEL] = iat_ransac('PTS1, PTS2, TRANSFORM')
Description: iat_ransac implements the ransac algorithm adapted to estimate the optimum TRANSFORM from point correspondences PTS1⟨−⟩PTS2 and returns the largest inlier set of correspondences and the respective transformation. [Cited from the iat toolbox]
2. [MAPPING, NUMMATCHES, INDEXA, INDEXB] = iat_match _features ('DESCA, DESCB, RATIO')
Description: iat_match_features implements descriptor matching between two sets of descriptors (features). The inverse cosine (arccosine) of the angle between two descriptors is used for their matching evaluation. Note that the function resolves double matches [Cited from the iat toolbox]

3 Motion Detection using frame differencing and background subtraction

Frame differencing is done by XORing particular set of images as by using bitxor function of MATLAB®. Background subtraction is the most important part of this whole stage because the main motion detection is performed by this. In background subtraction method we extract the required object features from the image, eliminating the background using a filter which in our case was Gaussian laplacian. We have tried to apply all the techniques which were used in the original system COCOA.

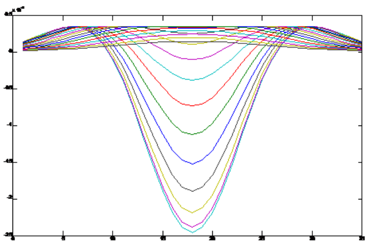


Figure 3: Response of the Gaussian filter.

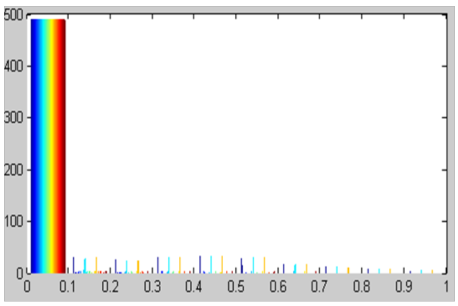


Figure 4: Histogram formed.

4 Moving object detection

In original project system there was an option of image tracking, but here we were not able to implement this function so we try to detect moving object using + sign. Technique use in detecting moving object was the blob

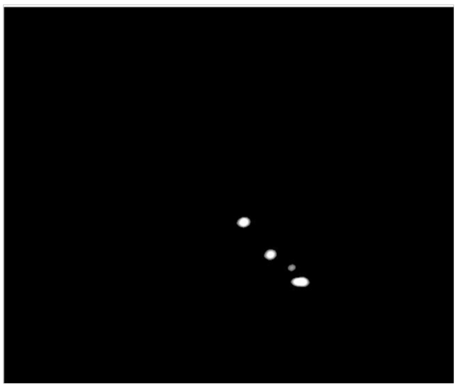


Figure 5: Image obtained by taking difference of pixel intensities.

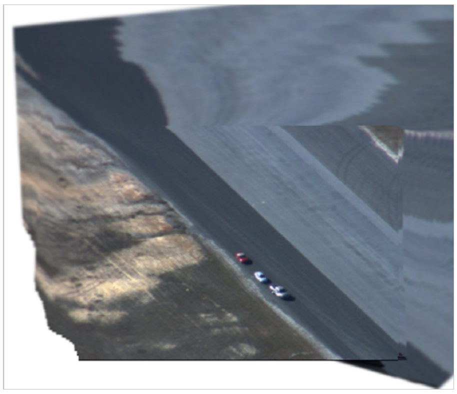
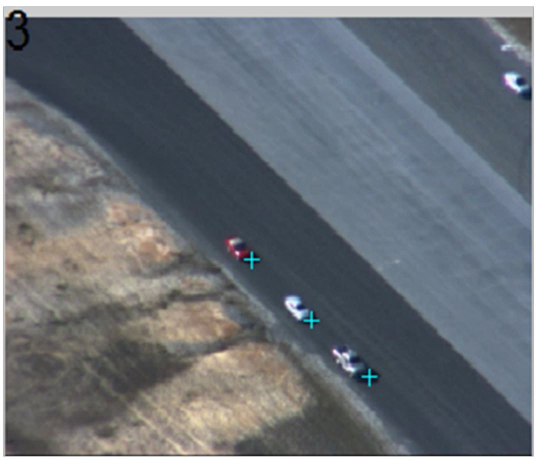


Figure 6: Mosaic of first stage.

detection technique, in which after convolving image with filter blobs were filtered out i.e. we chose a threshold that if a blob is greater than that, that particular blob is retained and rest of the part is blackened. The size of for blob in thresholding depends on the size of the object. Ideally adaptive filter should be made to used which can change its size according to the object but here we selected one optimal filter size for our test case as we had an idea of the size of moving objects in the frame. After thresholding the centroid of the blob was calculated which was done by imextrema function which finds the local maxima in the image. After assigning them the centroid we place a marker on that coordinate to represent that coordinate, and the number of coordinate formed are actually total number of moving objects. In our code the number of object detected were wrong in between in some frames due to wrong constant filter size. The shadow of the car was big enough to create another blob for itself leading to estimation of another moving object.



Function

1. [x,y,z,c]=imextrema(Image variable as input);
Description: Estimate extrema in an image at pixel resolution. The input image 'im1' is a grayscale image of any class. The outputs 'x' and 'y' specify the pixel positions of the extrema, 'z' specifies the value of the image at the extrema positions, and 'c' classifies the extrema, with:
c = -1 $\rightarrow$ local minimum
c = 0 $\rightarrow$ saddle point
c = +1 $\rightarrow$ local maximum
c = +2 $\rightarrow$ locally flat / extrema undefined
2. Thresholding: idx = find(blob_img $<$ 0.8);

References

[1] COCOA: Tracking in Aerial Imagery by Saad Ali and Mubark Shah

[2] Image alignment toolbox

[3] Student dave github for imextrem function and motion detection reference