Multi-UAV to UAV Detection and Tracking

Brandon Silva Waqas Sultani Mubarak Shah
University of Central Florida
brandon.silva209@knights.ucf.edu, waqas5163@gmail.com, shah@crcv.ucf.edu

Abstract

As UAVs (Unmanned Aerial Vehicles) are becoming more popular in personal, commercial, and government sectors, the need for an accurate and efficient system to detect small, fast moving objects (I.E other UAVs) is needed for collision avoidance and adjustment systems. Since UAVs can only carry a limited array of sensors, a camera is the inexpensive, lightweight choice that most UAVs are equipped with. Being able to utilize this visual data without other sensors is required to keep UAVs lightweight. In this paper, we propose a method for an end to end framework for detecting UAVs from the camera of another UAV. This is easily extendable for other small, fast moving objects (such as birds or projectiles) for collision avoidance, and further can be modified for detecting slower moving objects (such as animals or cars on the ground). Current optical flow methods are useful but expensive, and are subject to detecting false positives frequently. Utilizing 2D CNNs can achieve high accuracy rates with extremely low amounts of false positives. Our method can perform better than just using optical flow for detections, especially when there are many UAVs in the FOV of the camera.

1. Introduction

UAVs are becoming more easily available as the technology becomes cheaper, opening up new opportunities for many uses in a variety of sectors. To keep these UAVs light, compact, and cheap, inexpensive cameras offer many advantages as a sensor for collision avoidance on this platform.

For this optically based collision avoidance system, both spatial and temporal information must be utilized for accurate tracking of other UAVs. This information can be further processed to separate friendly and hostile objects, classifying objects, and calculating how far these objects are. This requires being able to process the video feed with as little latency as possible, along with processing on board the UAV, without requiring communication with other devices.

Video feed from these cameras presents quite a few challenges. (1) The video is taken from a moving camera that is mounted on the UAV, which introduces much noise from the camera shake as the UAV moves. (2) The UAVs are quite small against a noisy background (clouds, bright lights, ground buildings, trees etc), which introduces the main problem of detecting these small objects against a large FOV. (3) These UAVs must be detected at a considerable distance away such that the host UAV has time to perform evasion action. (4) There can be many UAVs at one time that require detection and avoidance, so it must be able to detect any number of UAVs in its FOV.

We propose a method that performs detections end to end, with the only stabilizing the input video before detection by our model. This proves to be an elegant, effective, and efficient system that requires less processing power than standard detection and tracking techniques.

2. Related Work

2.1. Optical Flow

The method introduced in J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and J. Li, D. HC. Bouman [2] utilized Background Motion Estimation using salient points and a global transformation for stabilization of the video. Then, they perform moving object detection by computing the background subtracted image given the estimation, compute local motion vector on the salient points, and then remove noise by comparing the local motion vectors to the background motion, where large differences indicate a detected UAV. For tracking, they use a Kalman filter to correlate UAVs between successive frames.

This method has a few issues we address in our method. First, UAVs are detected with only the context of the previous frame, where information about further frames is not utilized, which can provide better detections. Second, they utilize the full HD frame in their detections, which computing the necessary features (background subtracted image, global transformation) takes a lot of resources. This can be speed up considerably using smaller sized images.

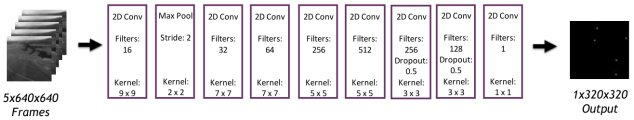

Figure 1: Proposed 2D CNN Model. Note the large kernel sizes used in the earlier layers, which give the filters more context for finding regions of motion. Batch Normalization is used after every convolutional layer, before the activation function is applied, except for the output layer.

2.2. Optical Flow with CNN

This method from D. H. Ye, J. Li, Q. Chen, J. Wachs, and C. Bouman. [3] uses the same method as described in [2], but uses a simple convolutional neural network that has a binary classifier, which has patches of motion passed to the model, where these patches are classified as either a true positive or a false positive. This removes false positives and gives better scores than just using optical flow, however it adds to the computation with little performance gain. Since it only detects on these patches, it only utilizes the spatial information of comparing true positives to false positives, and does not use temporal information, which limits it capability, and suffers similarly to the method above.

3. Proposed Method

Our proposed method uses a full 2D CNN that segments the input frames between detected UAVs and the background. The input video is stabilized to remove camera shake using a rigid Euclidean transformation based on keypoints matched between frames with the Lucas-Kanade method. This stabilization is smoothed over 30 frames to compensate for extreme camera shake often seen in many of the videos.

3.1. Stabilization

Stabilization is computed for all of the videos used for training and testing. Stabilization is performed as follows: (1) Find the transformation from the previous to current frame by using optical flow on all frames. This transformation is calculated using the Lucas-Kanade method for matching keypoints between frames. (2) Optical flow is computed from these matched keypoints, which gives the rigid transformation: $dx, dy, d\theta$ that specifies how much the frame needs to be transformed from the original frame. dx specifies the translation w.r.t the x axis, dy w.r.t the y axis, and $d\theta$ for rotation about the center of the frame. Given the previous keypoints and the current keypoints, this is done

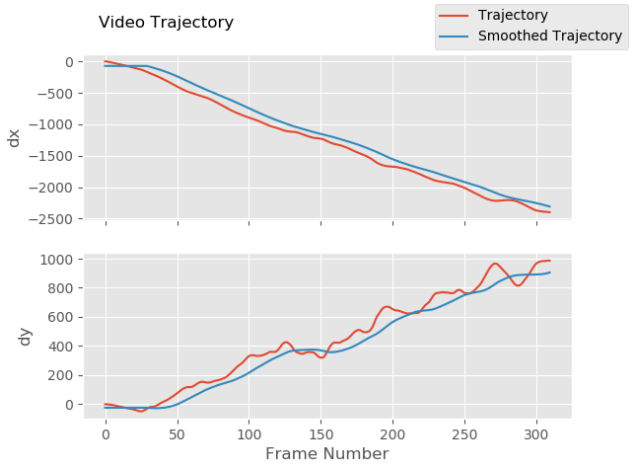

Figure 2: Trajectory vs Smoothed Trajectory for dx and dy for Clip 1

by solving the affine transformation matrix:

$$\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

Therefore, $ax + by + c = x_2$ and $dx + ey + f = y_2$
Where x, y is the previous point, x_2, y_2 is the current point, and a, b, c, d, e, f defines the affine transformation. Solving this requires at least 3 previous and current points and plugging in the above questions for x_2 and y_2, which gives the solution as:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{bmatrix}$$

Once the transform is computed, $dx, dy, d\theta$ is extracted from it as follows:

$$dx = c, dy = f, d\theta = arctan(\frac{d}{a})$$

(3) Smooth the trajectory by averaging the trajectory calculated in a given sliding window, (which we used 30 frames) to allow the stabilization to be more fluid between frames (see 2 for smoothed trajectory vs trajectory). (4) Update the transformations based on the smoothed trajectory: $transformation + (smoothed\_trajectory - trajectory)$. (5) Then, the transformation is applied to each frame.

3.2. Model

The proposed model is a full 2D convolutional neural network, which outputs a segmentation heatmap of detected

UAVs and the background (see 1). The model takes in 5 frames stacked along the channel axis: the middle frame and two consecutive frames before and after the middle frame. The segmentation is performed for the middle frame, so the model has temporal context for before and after the middle frame. The network has one max pooling layer, so the segmentation is half the size of the original input image. 2D convolutions are used instead of 3D to preserve the temporal relationship between frames, as convolutions are calculated across the entire volume of the 5 frames, as introduced in [1]. With this, we also propose splitting frames into 4 quadrants and performing detections on these quadrants. While this does increase computation time, it also provides a big accuracy increase as our model sometimes missed UAVs when there were many UAVs in a single frame. This can easily be modified to predict for the last frame, if being 2 frames behind is hindering fast detections. The model is extremely extendable, with the parameters shown in this paper empirically giving the best results, while only having around 6 million parameters.

4. Experiment Setup

4.1. Dataset

The dataset is from the authors of [2], which consists of 50 videos running at 30 FPS at a resolution of 1920x1080 and 1280x960, with a maximum of 8 UAVs in any given frame. There is 70250 frames in total. Annotations are given as bounding box coordinates, which are used to generate ground truth segmentation for training. These segmentations are generated using the center point of the ground truth bounding boxes, and filling a circle about the center point. This circle can either be binary (values of all 1's) or a Gaussian filter. Empirically, both label types perform the same, so Gaussian was used in our training and testing.

4.2. Model Training

The model is trained using MSE loss, where a constant of $k = 3$ is multiplied to the loss to weight UAV detections higher than the background. This is optimized using the Adam optimizer, with a learning rate of 0.00006, and a batch size of 24. We used 5 input frames into the network, where using the full frame was resized to 640x640, and the quadrants were resized to 540x540. The model was trained using 5-fold cross validation, where each fold consisted of 40 training videos and 10 testing videos. 2 Tesala V100 16GB GPUs were used to train the model.

5. Results

F1-Score is calculated if our predicated segmentations had overlap with the ground truth segmentations. Segmentations where counted as true positives if their center fell within 5 pixels of the ground truth center.

Method	F1-Score
Optical Flow [2]	0.77
Optical Flow w/ CNN [3]	0.806
Ours (Full Frame)	**0.7869**
Ours (Quadrants)	**0.8354**

Table 1: Results

(a) Full Frame

(b) Quadrants

Figure 3: Qualitative Results

Using the full frame, our model performed on par with the related works on this dataset. Using the quadrants of the frames significantly improved our results, outperforming the other methods proposed in [2] and [3] in detection, without using background subtraction or optical flow for detections. For the folds where test videos have frames with large number of UAVs in them, the quadrants method gave an 8% increase to F1-Score for those folds, and a 3-4% for the other folds.

The ridgeline and hills in the videos posed quite an issue with the network as it caused false detections in some of the videos, since all videos were recorded in the same area. Overall though, this end to end lightweight network for detecting these UAVs is shown to be a fast and efficient solution to this problem, without relying on optical flow or extensive preprocessing. This will further be enhanced with a novel tracking method to help reduce false alarms and increase the F1-Score.

References

[1] R. LaLonde, D. Zhang, and M. Shah. Clusternet: Detecting small objects in large scenesby exploiting spatio-temporal information, 2018. CVPR 2018. 3

[2] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman. Multi-target detection and tracking from a single

camera in unmanned aerial vehicles (uavs), 2016. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 1, 2, 3

[3] D. H. Ye, J. Li, Q. Chen, J. Wachs, and C. Bouman. Deep learning for moving object detection and tracking froma single camera in unmanned aerial vehicles (uavs), 2018. IST International Symposium on Electronic Imaging 2018. 2, 3