REPORT

TANMAY GARG(2018102021)

Objective

Our main objectives were:

- Ingestion of AOT dataset (Extending the Multi-Object Tracking to Airborne Objects).
- Integration of AOT with the Siamese Tracker.
- Obtaining Results (Accuracy, Low number of False Positives)
- Tracking Aerial Objects over Multiple Frames using the same object ID(number)

Problem Statement

Multi-Object Tracking : The goal of Online multi-object tracking is to estimate the spatio-temporal trajectories of multiple objects in an online video stream (i.e., the video is provided frame-by-frame). Aerial object tracking is a problem which is similar to multi object tracking. However, an aerial object's motion is much faster than the objects involved in trivial MOT datasets, making this problem much more challenging. The requirements of the desired solutions are; Very low number of false positives, Detection of the objects within the camera frame, Tracking of the objects to predict its future trajectory.

Placing the problem in larger context

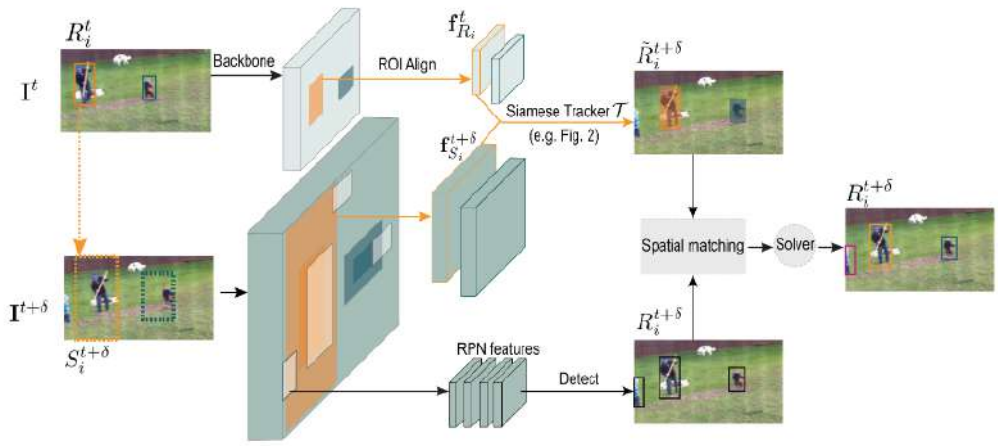
One of the key challenges in autonomous drone flight is that of the tasks to maintain enough distance from other airborne objects. While one can use several other sensors like lidar, electro-optical sensors etc, cameras are a very attractive alternative. This is because cameras are lightweight and are very rich in information. Aerial object navigation is a fundamental problem for numerous real-time applications, such as video surveillance, autonomous driving, and robot navigation. In this project we consider a solution that relies purely on information from the image sensor.

Flying airborne dataset presents a unique challenge. We do not just need to detect the obstacle but also to track it. Hence analysis of motion becomes very important when tracking aerial objects. Although we have focused on multi-object tracking in this project, the concepts learned are applicable and necessary for aerial object tracking.

Methodology Adopted

SiamMOT builds upon Faster-RCNN object detector which consists of a Region Proposal Network (RPN) and a region-based detection network. On top of the standard Faster-RCNN, SiamMOT adds a region-based Siamese tracker to model instance-level motion. As shown in Fig below SiamMOT takes as input two frames $\mathbf{I^t}$, $\mathbf{I^{t+\delta}}$ together with a set of detected instances $\mathbf{R^t} = \{R_1^t,...R_t^i,...\}$at time t. In SiamMOT, the detection network outputs a set of detected instances $\mathbf{R^{t+\delta}}$, while the tracker propagates $\mathbf{R^t}$ to time $t + \delta$ to generate $\acute{\mathbf{R}}^{\mathbf{t+\delta}}$.

SiamMOT contains a motion model that *tracks* each detected instance from time t to $t + \delta$ by propagating the bounding box R_i^t at time t to $\acute{R}_i^{t+\delta}$ at $t + \delta$; and a spatial matching process that *associates* the output of tracker $\acute{R}_i^{t+\delta}$ with the detections $R_i^{t+\delta}$ at time $t + \delta$ such that detected instances are linked from t to $t + \delta$.



1. Motion modelling with Siamese tracker

In SiamMOT, given a detected instance i at time t, the Siamese tracker searches for that particular instance at frame $\mathbf{I^{t+\delta}}$ in a contextual window around its location at frame $\mathbf{I^{t}}$. Formally,

$$(v_i^{t+\delta}, \acute{R}_i^{t+\delta}) = T(f_{Ri}{}^{t}, f_{Si}{}^{t+\delta}; \Theta)$$

where T is the learnable Siamese tracker with parameters Θ, $f_{Ri}{}^{t}$ is the feature map extracted over region R_i^t in frame $\mathbf{I^t}$, and $\mathbf{f_{Si}^{t+\delta}}$ is the feature map extracted over the search region $S_i^{t+\delta}$ in frame $\mathbf{I^{t+\delta}}$. We compute $S_i^{t+\delta}$ by expanding R_i^t by a factor r while maintaining the same geometric center as shown in Fig 1. We extract features $\mathbf{f_{Ri}^t}$ and $\mathbf{f_{Si}^{t+\delta}}$ using the region of interest aligned layer of Mask-RCNN. Finally, $v_i^{t+\delta}$ is the confidence of visibility for detected instance i at time $t + \delta$. In the context of multi-object tracking, we apply the above transformation multiple times, once for each detected instance $R_i^t \in \mathbf{R^t}$.

We conjecture that motion modelling is particularly important for online MOT. Specifically, association between R^t and $R^{t+\delta}$ will fail if

- $\acute{R}^{t+\delta}$ does not match to the right instance in $R^{t+\delta}$
- $v_i^{t+\delta}$ is low for a visible person at $t + \delta$.

2. Implicit motion model

Implicit motion model (IMM) uses an MLP to implicitly estimate the instance-level motion between two frames. In detail, the model concatenates $\mathbf{f_{Si}^t}$ and $\mathbf{f_{Si}^{t+\delta}}$ and feeds that to an MLP that predicts the visibility confidence v_i and the relative location and scale changes:

$$m_i = [\frac{x_i^{t+\delta} - x_i^t}{w_i^t}, \frac{y_i^{t+\delta} - y_i^t}{h_i^t}, \log\frac{w_i^{t+\delta}}{w_i^t} \log\frac{h_i^{t+\delta}}{h_i^t}]$$

in which (x_i^t, y_i^t, w_i^t, h_i^t) is the parameterization of R_i^t. We can trivially derive $\acute{R}^{t+\delta}$ from an inversion transformation of the above equation by taking as input R_i^t and m_i.

Loss. Given a triplet (R_i^t, $S_i^{t+\delta}$, $R_i^{t+\delta}$), we train IMM with the following training loss:

$$\mathbf{L} = \ell_{focal}(v_i, v_i^*) + \mathbb{1}[v_i^*]\,\ell_{reg}(m_i, m_i^*)$$

where v_i^* and m_i^* refer to ground truth values derived from $R_i^{t+\delta}$, 1 is the indicator function, ℓ_{focal} the focal loss for classification and ℓ_{reg} the commonly used smooth ℓ_1 loss for regression.

3. Explicit motion model

EMM uses a channel-wise cross-correlation operator (∗) to generate a pixel-level response map $\mathbf{r_i}$, which is effective in modelling dense optical flow estimation and in SOT for instance-level motion estimation. In SiamMOT, this operation correlates each location of the search feature map $\mathbf{f_{Si}}^{t+\delta}$ with the target feature map $\mathbf{f_{Ri}}^{t}$ to produce $\mathbf{ri} = \mathbf{f_{Si}}^{t+\delta} * \mathbf{f_{Ri}}^{t}$, so each map $\mathbf{r_i}$[k,:,:] captures a different aspect of similarity. EMM uses a fully convolutional network ψ to detect the matched instances in $\mathbf{r_i}$. Specifically, ψ predicts a dense visibility confidence map $\mathbf{v_i}$ indicating the likelihood of each pixel to contain the target object, and a dense location map $\mathbf{p_i}$ that encodes the offset from that location to the top-left and bottom-right bounding box corners. Thus, we can derive the instance region at (x,y) by the following transformation $R(\mathbf{p}(x,y)) = [x - l, y - t, x + r, y + b]$ in which $\mathbf{p}(x,y) = [l,t,r,b]$. Finally, we decode the maps as follows:

$$\tilde{R}_i^{t+\delta} = \mathcal{R}(\mathbf{p}_i(x^*, y^*)); \quad v_i^{t+\delta} = \mathbf{v}_i(x^*, y^*)$$
$$\mathbf{s.t.}(x^*, y^*) = \underset{x,y}{\operatorname{argmax}}(\mathbf{v}_i \odot \boldsymbol{\eta}_i)$$

where $\odot$ is the element-wise multiplication, $\boldsymbol{\eta}_\mathbf{i}$ is a penalty map that specifies a non-negative penalty score for the corresponding candidate region as follows:

$$\boldsymbol{\eta}_i(x, y) = \lambda \mathcal{C} + (1 - \lambda)\mathcal{S}(\mathcal{R}(\mathbf{p}(x, y)), R_i^t)$$

where λ is a weighting scalar ($0 \leq \lambda \leq 1$), C is the cosine window function w.r.t the geometric center of the previous target region R_i^t and S is a Guassian function w.r.t the relative scale changes between the candidate region ($\mathbf{p}(x,y)$)) and $\mathrm{R_i^t}$. The penalty map $\boldsymbol{\eta}_\mathbf{i}$ is introduced to discourage dramatic movements during the course of tracking.

Loss. Given a triplet (R_i^t, $S_i^{t+\delta}$, $R_i^{t+\delta}$), we formulate the training loss of EMM as follows:

$$\mathbf{L} = \sum_{x,y} \ell_{focal}(\mathbf{v}_i(x, y), \mathbf{v}_i^*(x, y)) + \sum_{x,y} \mathbb{1}[\mathbf{v}_i^*(x, y) = 1](w(x, y) \cdot \ell_{reg}(\mathbf{p}_i(x, y), \mathbf{p}_i^*(x, y)))$$

where (x,y) enumerates all the valid locations in $\mathrm{S_i^{t+\delta}}$, ℓ_{reg} is the IOU Loss for regression and ℓ_{focal} is the focal loss for classification. Finally, $\mathbf{v_i}^*$ and $\mathbf{p_i}^*$ are the pixel wise ground truth maps. $\mathrm{v_i}^*(x,y) = 1$ if (x,y) is within $R_i^{*t+\delta}$ and 0 otherwise. $\mathbf{p_i}^*(x,y) = [x - x_0^*, y - y_0^*, x_1^* - x, y_1^* - y]$ in which (x_0^*, y_0^*) and (x_1^*, y_1^*) corresponds to the coordinates of the top-left and the bottom-right corner of ground truth bounding box $R_i^{t+\delta}$.

We keep a trajectory active until it is unseen for $\tau = 30$ frames.

4. Using RESNET

a. Tested the Resnet – 50 architecture on CIFAR-10 dataset by varying the resolution of the images. To do this resolution was reduced by using two methods: global average pooling and image subsampling in whichever alternate row or col was left out depending on the desired resolution and performance in both cases was similar.

b. To test this we tested Resnet 50 on Amazon AOT dataset. To maintain it as a classification problem instead of an object detection problem we used the bounding boxes to take the image out which was used for classification. For testing purposes the size of the bounding boxes obtained using the labels in the dataset was

    I. Either kept same and the scaled to 224*224 and fed into the network.

    II. Box of size 224*224 where a disturbance was added to the center of the bounding box and then a box of size 224 *224 was taken and fed to the network in this case the entire object was often not in the image

Datasets

MOT17 is the most widely used multi-person tracking benchmark. It consists of 7 training and 7 test videos, ranging from from 7 to 90 seconds long. The videos feature crowded scenes in indoor shopping malls or outdoor streets. We report our results using several metrics: MOTA (Multiple Object Tracking Accuracy), IDF1 (ID F1 score), FP (False Positives), FN (False Negatives) and IDsw (ID switches).

The **CIFAR-10** dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

The **Airborne Object Tracking (AOT)** dataset is a collection of flight sequences collected onboard aerial vehicles with high-resolution cameras. To generate those sequences, two aircraft are equipped with sensors and fly *planned* encounters. The trajectories are designed to create a wide distribution of distances, closing velocities, and approach angles. In addition to the so-called planned aircraft, AOT also contains other unplanned airborne objects, which may be present in the sequences. Those objects are also labeled but their distance information is not available.

Results Obtained

1. Using Siamese Tracker

0:10

0:35

1:03

1:08

The above 4 screenshots are taken from the resultant video which was produced after detection of people from a video which was taken by police using a drone to catch people who were not following covid protocols.

Following link contains various video sequences (some from campus, aerial view): https://drive.google.com/drive/folders/1m5vBJZc9s0r4L0iSnLn8pqgdR-X8SDOc?usp=sharing

	num_frames	MT	PT	ML	IDs	FP	FN	MOTA	MOTP	IDF1
MOT17-02-DPM	600	8	26	28	42	44	11167	39.4%	0.087	43.5%
MOT17-02-FRCNN	600	10	30	22	47	44	10321	44.0%	0.093	50.3%
MOT17-02-SDP	600	12	34	16	83	55	8908	51.3%	0.093	50.8%

MOT17-04-DPM	1050	41	23	19	31	162	12818	72.6%	0.070	74.5%
MOT17-04-FRCNN	1050	41	25	17	24	177	14022	70.1%	0.071	73.2%
MOT17-04-SDP	1050	55	21	7	47	197	7378	84.0%	0.071	78.5%
MOT17-05-DPM	837	40	61	32	39	107	2358	63.8%	0.066	65.6%
MOT17-05-FRCNN	837	43	59	31	46	108	2462	62.2%	0.068	66.0%
MOT17-05-SDP	837	47	62	24	54	112	2017	68.4%	0.068	68.7%
MOT17-09-DPM	525	12	13	1	32	34	1658	67.6%	0.067	54.8%
MOT17-09-FRCNN	525	12	12	2	35	33	1820	64.5%	0.070	52.7%
MOT17-09-SDP	525	14	12	0	35	35	1582	69.0%	0.067	53.5%
MOT17-10-DPM	654	22	27	8	92	144	4099	66.2%	0.132	57.6%
MOT17-10-FRCNN	654	31	24	2	134	204	3280	71.8%	0.137	61.1%
MOT17-10-SDP	654	32	24	1	162	263	2647	76.1%	0.139	62.4%
MOT17-11-DPM	900	25	27	23	13	87	2887	68.3%	0.057	71.1%
MOT17-11-FRCNN	900	31	27	17	12	96	2408	73.3%	0.061	74.9%
MOT17-11-SDP	900	43	18	14	21	97	1807	79.6%	0.064	77.7%
MOT17-13-DPM	750	35	46	29	51	56	5006	56.1%	0.110	61.5%
MOT17-13-FRCNN	750	63	37	10	96	91	2431	77.5%	0.116	75.3%
MOT17-13-SDP	750	71	23	16	90	99	2480	77.1%	0.118	74.8%
OVERALL	15948	688	631	319	1186	2245	103556	69.6%	0.084	67.7%

Detailed result summary on MOT17 test videos.

τ (frames)	**MOTA**	**IDF1**
1	63.8	62.1
5	63.9	61.8
10	65.2	64.3

30	67.8	66.2
50	69.6	67.7

Results of SiamMOT inference that terminates active trajectories after they are unseen within τ consecutive frames.

Results in the above Table shows that tracking performance increases with τ, especially IDF1 score that measures the temporal consistency of trajectories. This means that our tracker is capable of tracking beyond a few consecutive frames.

2. Using RESNET

a. Tested the Resnet – 50 architecture on CIFAR-10 dataset

Resolution (Average pooling)	Accuracy
1x	95%
.5x	86.5%
.25x	73.4%
.125x	53.4%

Resolution (subsampling)	Accuracy
1x	98%
.5x	87.5%
.25x	71.4%
.125x	51.4%

The results clearly show a massive drop in performance as the resolution of the image goes down.

b. Tested the Resnet – 50 architecture on Amazon AOT dataset

Size of bounding box	Accuracy

224*224	88%
Distorted 224*224	76%
448*448	69%

In this case as well the accuracy was decreasing rapidly for as size of the drone wrt to image decreased.

Codebase

We took the SiamMOT codebase from Amazon Research (link). We made several changes to the existing codebase to counter the various GPUs, nvidia and cuda toolkit versions. The apex and cocoapi needed to have a different environment setup. We further made modifications to encounter the AOT problem. The updated codebase is siam-mot .

Conclusion

We studied research papers on JDE (JDE with RCNN, JDE with Graphical Neural Network) and a paper on SiamMOT to achieve better understanding of Multi-Object Tracking and finally we chose to proceed with SiamMOT as it uses a region-based MOT network – SiamMOT, which detects and associates object instances simultaneously. In SiamMOT, detected instances are temporally linked by a Siamese tracker that models instance motion across frames. Its framework can be easily adapted to accommodate multi-class multi-object tracking. Further we also used RESNET-50 architecture on the CIFAR-10 dataset and the Amazon AOT dataset. We implemented the average pooling and sampling technique. We have discussed the methodology above.

We successfully imported the codebase with several required changes (removed all hardware and software limitations as well as the different environment clashes) in it. While implementing the above methodology we ingested and integrated the AOT dataset with the Siamese Model. The changes and the link to the updated codebase is shared above.

We not only achieved the desired results but also improved it by constant training and testing.