Using Temporal and Spatial Features to Track Fast Moving Objects

Kristi Bushman, Clara Cardoso Ferreira, Avery Peiffer

February 25, 2021

1 Introduction and Goal

Fast Moving Objects (FMOs) are objects that move with a velocity greater than that which can be captured by the exposure rate of a camera [1]. FMOs appear in images as long streaks that can be many times larger than their actual size, producing a blurring effect. These objects present a challenge for object tracking, since traditional object tracking methods assume relatively small variance between the ground truth appearance of an object and its representation while moving. Since real-time tracking is especially useful in the physical world, it must be robust to noise, occlusions and blurring in addition to being fast. Thus far, real-time tracking methods for FMOs rely on the use of convolutional neural networks (CNNs), which focus on visual feature information to estimate the object's position. The accuracy and robustness of FMO tracking methods can increase by incorporating contextual temporal information into tracking computations, which can be accomplished with the Recurrent Convolutional Neural Network (CRNN) architecture.

Our method will use CRNNs to track FMOs so that our application not only extracts visual features but also models frames sequentially. It is a combination of convolutional neural network (CNN) architecture and recurrent neural network (RNN) architecture [2]. RNNs are used to detect patterns in which surrounding measurements (in time or space) have a bearing on current measurements [3]. A common use for RNNs is in auto-correct typing features: when a spelling mistake is made at the end of a word, it is necessary to incorporate the previous characters to correctly estimate the intended word. The ability of RNNs to use previous information to predict the next set is the general idea that we wish to apply to FMO tracking; the location of the object in previous frames should prove useful in calculating its location in the current frame. Time permitting, we may try to extend our method to be able to predict the location of the FMOs at later timestamps. Again, this is based on the idea that the trajectory in previous frames may be useful for determining the objects location at a later time.

2 Importance

With the rapid pace of technological development in machine learning, computer vision and hardware sensors, it is becoming more convenient and safe for machines to operate autonomously. To this end, autonomous tracking systems could eliminate the need for mentally taxing and monotonous tasks such as surveillance, navigation, terrain exploration, sports monitoring and analytics, and other tasks that require fast tracking [4]. Real-time tracking of fast moving objects plays an important role in robotic decision making to avoid moving obstacles, map and localize robots, navigate high-speed vehicles, and estimate poses [5]. More compromising applications can arise in military command and control, facility security, and emergency services such as policing and fire-fighting [6]. Situational awareness in space is also a motivation to track fast moving objects, since small but fast objects can cause damage to space navigation systems [7]. The wide variety of FMO tracking applications and the growing demand for automation indicates that more accurate, robust, and faster tracking methods are necessary.

3 Challenges

Tracking objects outside the lab environment is a challenge in itself because the physical world is complex with occlusions, noise, and unique environments. Furthermore, predicting the trajectory of a tracked object

presents an even greater difficulty since objects of varying elasticity can bounce on surfaces with unique appearance and depth.

It is challenging to use machine learning to track FMOs because the speed of the object can cause it to appear distorted in the video frames. Additionally, the fast motion causes the position of the object to change greatly from one frame to the next, increasing the possibility of tracking errors. Most object recognition is done by learning features of the object's appearance, then detecting the presence of those features in the video frames. However, this is a challenge with FMOs because the objects of interest often look different in each frame. A fast moving object can appear to change size and shape, depending on the lighting and speed of the object [1]. For example, a volleyball at rest will appear round with colored stripes. However, when the volleyball is moving fast and rotating, it may appear as a single color streak that blends in with the background. Because the features are not consistent from frame to frame, it can be difficult to detect the object. Furthermore, many object tracking methods use spatial locality to help locate the object based on its position in previous frames. With slow moving objects, the position of the object will usually only change by a few pixels from frame to frame. However, an FMO's position may change greatly between frames. This may make it difficult to correctly identify the object in consecutive frames.

Our method must to learn to use both the appearance and the locality of an FMO in order to track FMOs. It must be robust to FMOs' varying characteristics between frames, so it should depend on more than a single set of features, space and time.

4 Related Research

This section introduces previous work that has been done on object tracking, recurrent neural networks, and FMOs. Section 4.1 focuses on commonly used tracking methods, while section 4.2 examines the use of RNNs for tracking. Model-based tracking of FMOs is discussed in section 4.3. Section 4.4. informs existing learning-based methods for FMO tracking using conventional cameras and event-based cameras. Section 4.5 focuses on trajectory prediction methods.

4.1 Approaches to Object Tracking

4.1.1 Traditional Methods For Object Tracking

A common optical flow method for tracking is the Lucas Kanade algorithm [8] and its variations. It assumes that the moving object is similar to and is close to the surrounding pixels in the following frame. It obtains the gradients from comparing the moving object to a template. It uses the target's neighboring pixels to compute the Hessian matrix, which gives the the position of the target. However, this algorithm assumes the target maintains constant flow and that the illumination of the target does not vary widely, which is unreasonable for tracking multiple frames. Furthermore, it is not robust to occlusions, shadows and objects that change shape across its traejectory [9].

Traditional approaches to object tracking estimate an object's position by incorporating the current frame into a running estimate built from previous frames. The single-constraint-at-a-time (SCAAT) approach, in which tracking reports are produced each time the object's position is measured. By applying current tracking reports to previous estimates, SCAAT's estimation of an object's position improves incrementally with very low latency. However, SCAAT works most accurately in combination with other tracking algorithms such as the Kalman filter [10], which is not robust to FMOs.

Tracking can also be achieved by combining scale-invariant feature transform (SIFT) feature tracking with the mean-shift algorithm to track specific regions of interest in an image. This method is generally more robust to occlusion, as it relies on a maximum-likelihood estimate of the respective measurements from mean shift and SIFT. Both of these methods assume the object to be tracked is moving relatively slowly through the frame, in that a definitive estimate can be made of its location [11]. The blur caused by FMOs presents a different challenge by introducing more uncertainty into the object recognition step.

4.1.2 Machine Learning Methods For Object Tracking

The Simple Online Realtime Tracker extended with a deep association metric, also known as Deep SORT, is the most popular method to track objects in real time due to its robustness to occlusion while maintaining performance [12]. SORT uses the Kalman Filter to detect the object of interest, followed by the Hungarian algorithm to estimate the object's position in the following frame. A CNN is used to detect spatial features of the object, making it robust to occlusion and improving its tracking capabilities.

Deep SORT tracking is limited in highly complex real-world environments, where multiple similar objects can confuse the algorithm. The authors of [13] compute predictions' confidence for the next frame and eliminate predictions with low confidence to prevent confusion. They focus on multi-object tracking of vehicles in busy traffic rather than fast moving objects with a wide range of trajectories. By removing low confidence tracking instances, this method may result in lost information. Furthermore, this method may not work as well when applied to objects moving at higher speeds.

4.2 Tracking with Spatial Temporal Neural Network

Although RNNs have not yet been applied to tracking of FMOs, several papers have outlined the use of RNNs for general object tracking. This section along with sections 4.4.2 and 4.5 uses these papers as inspiration for our research.

The ROLO (recurrent YOLO, meaning "you only look once") method is an RNN that was developed for object tracking (using bounding boxes). First, the method uses the YOLO deep convolutional neural network to detect visual features. Next, fully-connected layers in the network use the feature information to make preliminary predictions about object locations. Finally, both the raw feature information and the preliminary location predictions are fed into a recurrent long short-term memory (LSTM) network. The LSTM incorporates information from the previous frames in order to help make the prediction of object's location in the current frame. The ROLO method performs well on baselines. Because the method incorporates temporal information, it successfully detects objects in the presence of blur and occlusions [14]. YOLO is fast since it divides the frame into a grid to find the most probable grid with a moving target [15]. The ROLO could potentially be adapted to track small objects by using fine-YOLO which is specific to small objects [16].

We chose to use Recurrent Network for Multiple Object Video Object Segmentation (RVOS), which is also a recurrent convolutional network [17]; it performs tracking using segmentation rather than bounding boxes. Similar to the ROLO network, the RVOS network takes advantage of both the visual features and the temporal continuity of the object over sequential frames. Additionally, this network is able to distinguish different instances when there are multiple objects in the frame. It does this by using recurrence in the spatial dimension in addition to the temporal dimension. The neural network has an encoder-decoder structure which allows it to make pixel-wise predictions for segmentation.

4.3 FMO-Dedicated Model-based Tracking Methods

Rozumnyi et al. [1] identified the need for a dataset dedicated to FMOs, as other existing object tracking datasets (ALOV, OTB, and VOT) did not contain a good representation of these types of objects. They created and annotated the FMOv2 dataset so that they could work on the FMO tracking problem.

Their first attempt at localizing FMOs was the FMOd method which consisted of a series of three algorithms: the detector, the re-detector and the tracker [1]. First, the detector algorithm locates the FMO based on clear contrast with the background in three consecutive frames. The detector also estimates properties of the FMO such as color and radius. Next, the re-detector algorithm uses information from the detector to try to locate the FMO in the previous and successive frames. The re-detector only looks for the object in portions of the image where the object would be likely to appear based on its previous motion. The re-detector is less sensitive to blur, occlusions, or blending with the background because it already has information about the likely location of the object. If the detector and re-detector fail, then the tracker algorithm is used. The tracker is an image synthesis technique that incorporates information about motion, color, and radius of the FMO in order to locate it.

Kotera and Rozumnyi et al. extended the TbD method to create a new method called Non-Causal TbD (NC-TbD). Unlike the original TbD algorithm, NC-TbD does not assume that the trajectory in the current

frame is causally related to the previous frames. This is important in cases when, say, a ball changes direction due to contact with the ground or a player. The method uses dynamic programming to detect changes in motion (in this case, bounces) [18].

4.4 Learning-based Methods for Tracking FMOs

4.4.1 FMO Tracking Using Conventional Camera Data

There are a few existing learning-based methods for tracking FMOs. These methods were introduced in order to address some of the problems with traditional model-based methods. The model-based methods are all attempting to disentangle and solve several related tasks (deblurring, matting, tracking, trajectory estimation, appearance recovery, etc.). This results in a very complicated model. Each of these models have several parameters that need tuning and operate based on several assumptions (for example, a static background) which may not always hold. Additionally, these methods are slow, taking up to four seconds per frame. Learning-based methods were introduced because they can learn without needing to disentangle and understand the complexities of each of the different tasks. They can also perform inference significantly faster.

FMODetect was the first learning based method proposed by Rozumnyi et al. The method first uses a convolutional neural network to detect the FMOs. The output from the detection network is then processed and fed into a "matting and fitting" network. This network has an encoder-decoder structure that helps to separate the FMO from the background. Finally, a deblurring step is applied. This learning-based method outperformed the TbD method [19].

Zita et al. also propose a learning-based approach for FMO tracking. They tried several different existing networks for segmentation and found the best results with ENet [20]. The results are better than the method from [1] and the inference time is significantly faster.

DeFMO is the most recent learning-based method for FMOs proposed by Rozumnyi et al., although with a slightly different goal. This method used a neural network that takes in a single frame and its estimated background, and outputs a series of sub-frames (e.g. what would have been captured with a high-speed camera). They compared their results to the ground-truth frames of the same sequence captured by a high-speed camera. The results were very positive [21].

4.4.2 High-speed Tracking Using Event-based Data

With the development of high-speed event-based cameras in recent decades, it is worth mentioning high-speed tracking trade-offs between conventional cameras and event-based technologies, also known as neuromorphic cameras. Neuromorphic cameras have microsecond latency, while conventional cameras have milisecond latency. The biologically inspired hardware of event-based cameras can enable much more data capture given a specified period of time. Therefore, general characteristics observed in FMOs, such as distortion, do not affect event-based data.

Event-based sensors analyze motion rather than independent frames, so RNNs have been used for a variation of event-based analysis. RNNs analyze the relationship between multiple consecutive instances [22], which can enable both tracking capability and depth estimation [23]. A potential benefit of using RNN tracking with conventional cameras over tracking with neuromorphic cameras is that conventional cameras may gather spatial clues from the environment, whereas neuromorphic cameras do not capture the idle background. The benefit of tracking FMOs with a neuromorphic camera is that blurring, shape distortion and lighting is much more easily avoidable due to its low latency and high dynamic range. Aside from the draw of neuromorphic technology's low power consumption [22], this research can provide more information on traditional tracking capabilities for further comparison.

Prior neuromorphic RNN tracking research using a DAVIS camera and a Loihi processor, which provide event-driven vision and processing, respectively, has shown to perform with significant speed. Another neuromorphic tracking method was analyzed and showed that although neuromorphic cameras track more accurately than conventional cameras [24], they are not widely used. A trade-off between tracking accuracy and frame rate must be made for each application, since the optimal frame rate required varies inversely [25]. Therefore, researching how to improve conventional (low to medium frame speed) tracking methods can provide more application-specific options for FMO tracking.

4.5 Using Trajectory To Predict Object Location in Later Frames

Trajectory estimation is a topic of interest in many fields including autonomous vehicles [26] and drones [27]. In both cases, RNNs and LSTMs have been used to predict motion. The KITTI dataset has been used to estimate the trajectory of objects in the vehicle view. This dataset provides not only visual data, but also states of the vehicle such as yaw rate, velocities and accelerations of the vehicle. Two seconds of these inputs are fed into a deep neural network to estimate the trajectory of the following two seconds [26]. For the UAV collision avoidance trajectory estimation research, short-term prediction, with a set of frames and weights, and a threshold are fed into the neural network. The output prediction is an ellipsoid shaped area of pixels where the tracked obstacle is expected to be [27]. Other related research on trajectory prediction also uses RNNs and LSTMs for pedestrian traffic [28]. Additional ball trajectory prediction use two ANNs: one for the first bounce and another for the second bounce [29]. These trajectory estimation methods can aid the development of a network to predict trajectories of FMOs.

5 Method

5.1 High-level Idea

Our plan is to use a recurrent convolutional neural network for segmentation on the FMO dataset, more specifically RVOS. The idea is that the convolutional aspect of the network will be able to detect the FMO when the visual features are clear. The recurrent nature of the network will help in cases where there is severe blurring and blending with the background. The previous frames will give the network some idea of where the object is expected to be in successive frames.

We will be using existing code for the RVOS network that was provided by the authors of [17]. Although we hope that the RVOS network will be able to track FMOs without many modifications, we anticipate that that the challenges posed by fast moving objects will require many modifications such as deblurring and changing the loss function. We may try several modifications to optimize FMO tracking and compare accuracy and speed tradeoff of RVOS.

Time-permitting, we may also extend our method to predict future locations of an FMO. The general idea is if the network can identify the location of an FMO in several consecutive frames, then it may be able to determine its trajectory and predict the object's location at a future time (say, 10 or 20 frames ahead). In order for this method to work, the network must develop an understanding of other objects in the scene and how they might affect the trajectory. For example, it must learn that when a ball is hit upwards, gravity will eventually bring it back down. It must learn that when a ball comes into contact with a player, it may change direction. We anticipate this to be a challenge, but feel it is an interesting idea to try out if we have time.

5.2 Novelty

To the best of our knowledge, there have not been any attempts at using a recurrent convolutional neural network for FMO tracking. Although the traditional model-based methods for FMO tracking incorporate information from previous frames, all of the existing learning-based methods (CNN) operate on a single frame at a time. The traditional model-based methods have shown that the temporal information can be useful for tracking the object, so we believe it is likely that the temporal information can help to improve the learning-based methods as well.

To the best of our knowledge, the prediction of the location of an FMO at a later time has not been done before. However, motion estimation has been done with videos, such as movement of people [30] and cars [26].

6 FMOv2 Dataset and Dataset Generator

The primary dataset that we plan to use is the FMOv2 dataset that was created by Rozumnyi et al [1]. This dataset consists of several videos of fast moving objects in sports (volleyball, tennis, ping pong, etc.).

There are nineteen different videos, each comprised of 50-100 frames. These still frames exhibit the semi-transparent streaks associated with FMOs and are shown in Figure 1. Our method will likely only use the previous three frames as context; thus, each set of four frames can be used as a training example. The dataset contains ground truth segmentation masks that show where the ball is located in each frame.

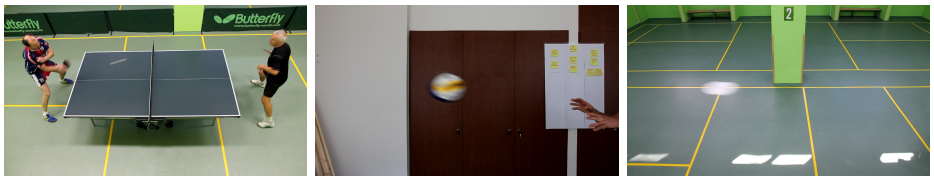


Figure 1: Sample images from FMOv2 dataset

We plan to supplement the FMOv2 dataset with a dataset generator to support the training of our network as Ales Zita and Filip Sroubek did in "Learning-based Tracking of Fast Moving Objects." This generated dataset will be labeled with ground truth and free from false FMOs (objects that appear to be FMOs but are not tracked) [20].

It would be interesting to compare our findings with similar datasets available, such as TbD-3D, TbD, Falling Objects; they can be found on FMO resources website.

An alternate dataset could be used in case the challenge of FMOs complicate training. We could pre-train our model using a non-blurred motion dataset, then fine-tune the model on the FMOv2 dataset. One possible dataset that could be used for pre-training is the Visual Object Tracking (VOT) dataset [31].

7 Metrics and Baselines

To evaluate our method, we will compare our results to both the traditional model-based methods (FMO [1], TbD [18]) and the learning-based methods (FMODetect [19], ENet [20]). We will use the metrics that were used in each of those papers in order to make our comparisons.

The most commonly used metric for this task is Trajectory Intersection over Union (TIoU). Intersection over union (IoU) is the number of pixels that are labeled positive in both the predicted and ground truth segmentation masks divided by the number of pixels that are labeled positive in either the ground truth or the prediction. TIoU is calculated by averaging the IoU values of all frames in the video sequence. Other metrics that have been used in this field are precision, recall, and F1 score [32].

$$TIoU = \frac{1}{|frames|} \sum_{f \in frames} \frac{TP_f}{TP_f + FP_f + FN_f}$$

$$Precision = \frac{1}{|frames|} \sum_{f \in frames} \frac{TP_f}{TP_f + FP_f}$$

$$Recall = \frac{1}{|frames|} \sum_{f \in frames} \frac{TP_f}{TP_f + FN_f}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

We will also be comparing the inference time for our algorithm against the other models. Ideally, the frames per second (fps) will be less than or equal to the frame rate of the video. This would mean that the algorithm can track in real time.

8 Projected Schedules

The projected schedules are listed below in table form for brevity. They are also attached as more comprehensive Gantt charts in the appendix; we will refer to these schedule throughout the course of the project.

8.1 Conservative Schedule

Task Group	**Task**	**Projected Completion Date**
Implement existing RVOS methods	Choose runtime environment	3/3
	Download dependencies	3/5
	Switch dataset to FMO	3/11
	Adapt RVOS to new dataset	3/27
Deblurring	Learn deblurring methods	3/20
	Combine deblurring methods to RVOS	3/26
	Overlay history of tracker	3/31
	Segmentation	4/2
Analyze deblurring	Intersection over union	4/1
	Document deblurring results	4/8
	Compare deblurring to existing methods	4/9
Final presentation	Synthesize results into presentation	4/13
	Review slides and practice presentation	4/15
Class deadlines	Homework 2	3/18
	Mid-semester report	3/25
	Homework 3	4/13
	Final presentation	4/22

8.2 Ambitious Schedule

Task Group	Task	Projected Completion Date
Implement existing RVOS methods	Choose runtime environment	3/3
	Download dependencies	3/5
	Switch dataset to FMO	3/11
	Adapt RVOS to new dataset	3/21
Deblurring	Learn deblurring methods	3/14
	Combine deblurring methods to RVOS	3/22
Analyze deblurring	Segmentation	3/17
	Intersection over union	3/21
	Document deblurring results	3/24
	Compare deblurring to existing methods	3/24
Trajectory planning	Display history of trajectory	3/28
	Repeat tracking multiple times per frame	4/2
	Predict trajectory	4/6
	Display trajectory prediction	4/8
	Compare trajectory to existing papers	4/9
Final presentation	Synthesize results into presentation	4/13
	Review slides and practice presentation	4/15
Class deadlines	Homework 2	3/18
	Mid-semester report	3/25
	Homework 3	4/13
	Final presentation	4/22

References

[1] Denys Rozumnyi, Jan Kotera, Filip Sroubek, Lukas Novotny, and Jiri Matas. The world of fast moving objects. 07 2017.

[2] Chao Duan, Steffen Junginger, Jiahao Huang, Kairong Jin, and Kerstin Thurow. Deep Learning for Visual SLAM in Transportation Robotics: A review. *Transportation Safety and Environment*, 1(3):177–184, 01 2020.

[3] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019.

[4] Daniel Gordon, Ali Farhadi, and Dieter Fox. Re3 : Real-time recurrent regression networks for visual tracking of generic objects, 2018.

[5] Boyoon Jung and Gaurav Sukhatme. Real-time motion tracking from a mobile robot. *I. J. Social Robotics*, 2:63–78, 03 2010.

[6] Nikhil Naikal. Towards autonomous situation awareness. page 1, 05 2014.

[7] B. A. Jones, D. S. Bryant, B. Vo, and B. Vo. Challenges of multi-target tracking for space situational awareness. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1278–1285, 2015.

[8] Bruce D. Luxas and Takeo Kanade. An iterative image registration technique with application to stereo vision, 1981.

[9] Shaul Oron, Aharon Bar-Hille, and Shai Avidan. Extended lucas-kanade tracking. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 142–156, Cham, 2014. Springer International Publishing.

[10] Vernon Reader and Gregory Welch. Scaat: Incremental tracking with incomplete information. 06 2001.

[11] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using sift features and mean shift. *Computer Vision and Image Understanding*, 113(3):345–352, 2009. Special Issue on Video Analysis.

[12] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric, 2017.

[13] X. Hou, Y. Wang, and L. Chau. Vehicle tracking using deep sort with low confidence track filtering. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2019.

[14] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017.

[15] L. Tan, X. Dong, Y. Ma, and C. Yu. A multiple object tracking algorithm based on yolo detection. In *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5, 2018.

[16] Minh-Tan Pham, Luc Courtrai, Chloé Friguet, Sébastien Lefèvre, and Alexandre Baussard. Yolo-fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing*, 12(15), 2020.

[17] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro i Nieto. Rvos: End-to-end recurrent network for video object segmentation, 2019.

[18] Jan Kotera, Denys Rozumnyi, Filip Sroubek, and Jiri Matas. Intra-frame object tracking by deblatting. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct 2019.

[19] Denys Rozumnyi, Jiri Matas, Filip Sroubek, Marc Pollefeys, and Martin R. Oswald. Fmodetect: Robust detection and trajectory estimation of fast moving objects, 2020.

[20] Ales Zita and Filip Sroubek. Learning-based tracking of fast moving objects, 2020.

[21] Denys Rozumnyi, Martin R. Oswald, Vittorio Ferrari, Jiri Matas, and Marc Pollefeys. Defmo: Deblurring and shape recovery of fast moving objects, 2020.

[22] A. Renner, M. Evanusa, G. Orchard, and Y. Sandamirskaya. Event-based attention and tracking on neuromorphic hardware. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 132–132, 2020.

[23] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events, 2020.

[24] Alpha Renner, Matthew Evanusa, and Yulia Sandamirskaya. Event-based attention and tracking on neuromorphic hardware. *CoRR*, abs/1907.04060, 2019.

[25] A. Mohan, A. S. Kaseb, K. W. Gauen, Y. Lu, A. R. Reibman, and T. J. Hacker. Determining the necessary frame rate of video data for object tracking under accuracy constraints. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 368–371, 2018.

[26] Jaskaran Virdi. Using deep learning to predict obstacle trajectories for collision avoidance in autonomous vehicles. In *ProQuest Dissertations Publishing*, pages 1278–1285, 2018.

[27] Vincent Kurtz and Hai Lin. Toward verifiable real-time obstacle motion prediction for dynamic collision avoidance, 2019.

[28] Anton Milan, Seyed Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks, 2016.

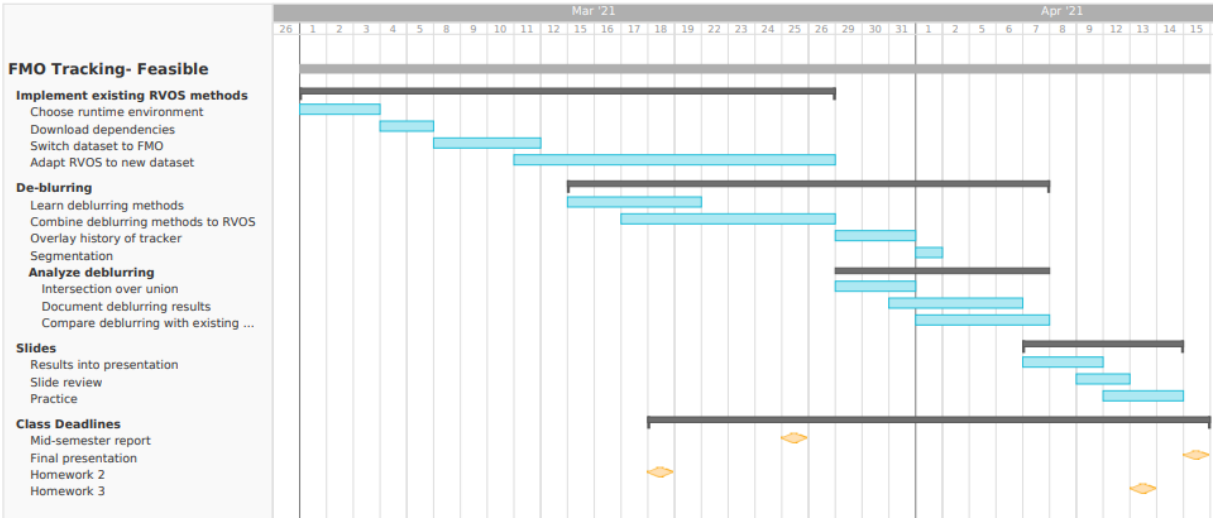
[29] H. Lin and Y. Huang. Ball trajectory tracking and prediction for a ping-pong robot. In *2019 9th International Conference on Information Science and Technology (ICIST)*, pages 222–227, 2019.

[30] Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A review on deep learning techniques for video prediction, 2020.

[31] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Čehovin Zajc, Martin Danelljan, Alan Lukezic, Ondrej Drbohlav, Linbo He, Yushan Zhang, Song Yan, Jinyu Yang, Gustavo Fernandez, and et al. The eighth visual object tracking vot2020 challenge results, 2020.

[32] Jan Kotera, Denys Rozumnyi, Filip Sroubek, and Jiri Matas. Intra-frame object tracking by deblatting. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct 2019.

Appendix A Conservative schedule Gantt chart



Appendix B Ambitious schedule Gantt chart