Multi-Hypothesis Extended Kalman Filter for Semantic Tracking*

*Omoruyi Atekha
Department of Mechanical Engineering
Stanford University
Stanford, USA
oatekha@stanford.edu

***Abstract*—Open Vocabulary Segmentation is a method that involves dividing an image into distinct sections utilizing a set of open semantic classes; this implies that the class categories are not strictly tied to the training data. While semantic segmentation allows for a wide range of applications in robotics, 3D modeling, and object tracking, it can produce inconsistent results across similar perspectives of a scene, making it less suitable for tasks requiring multi-view, continuous, or 3D perspectives. This inconsistency consequently hampers these models' capacity to efficiently track and localize objects within scenes without resorting to supplementary techniques. Addressing this limitation, our project proposes using a Multi Hypothesis Extended Kalman Filter (MHEKF) to bolster image segmentation's accuracy and reliability, consequently improving object detection and tracking. The system utilizes noisy semantic segmentation masks as bounding box pose measurements. The filter assumes a position-velocity dynamics model to predict the pose of a semantically segmented object in a collection of frames of a given scene. Qualitatively the system is able to improve tracking of unique semantic classes, and provides smoother results than the raw measurements.**

***Index Terms*—Kalman Filter, Object Detection, Object Tracking, Visual Transformer, Semantic Segmentation**

I. Introduction

The application of machine learning techniques in computer vision has resulted in a variety of models, algorithms, and technologies that have built on top of classical geometric techniques, such as photogrammetry, feature detection, optical flow, etc. These developments, which have facilitated the creation of 2D representations that were previously impossible to achieve with traditional techniques, and Motion Based Object Segmentation, which has led to more robust estimates for motion fields, scene flows, etc. Despite the increase in multi view perspective models, they remain unable to incorporate semantic understanding and segmentation into the scenes they produce. Typical scene extraction and object segmentation in multi view environments are feature and geometrically based, segmenting a given scene based on learned structural or visual cues [3] for a closed number of classes. These techniques typically require dense view annotations to capture the semantic representations, which are difficult to create and articulate, making these data sets rare. Furthermore, the current limitations of multi view semantic segmentation models are due to a lack of data linking complex multi view scenes to language, resulting in restricted categorical understanding [4]. Recent developments have enabled performative open-vocabulary segmentation within the 2D single view image space, using visual transformers that can create pixel-level segmentation masks for a collection of classes with methods like CLIPSEG, Segment Anything Model (SAM), and Open Vocabulary Segmentation (OV-Seg) [3] [5] [6]. While these open-vocabulary segmentation models may be performative under ideal conditions, their ability to accurately segment masks can be easily affected by occlusions or changes in perspective. Unlike other segmentation algorithms, open vocabulary segmentation models continually search for the most similar match at every pose, which can change depending on perspective or ambiguity of the open vocabulary class. Traditional object tracking methods may not be the ideal choice when the goal is to precisely track a specific object within a scene. While techniques like dense optical flow have proven effective, they often rely on categorically trained Convolutional Neural Networks (CNNs) limiting their general applicability. Semantically driven object tracking, on the other hand, opens up new possibilities, allowing robots to navigate and localize themselves in zero-shot scenarios by recognizing previously unseen objects.

Complex algorithms like SEEM from, "Segment Everything Everywhere All at Once" [2] offer a learned approach to predict the object's segmentation mask across frames. However, these methods demand substantial computational resources and are challenging to execute in real-time scenarios.One potential approach to tracking a given object, despite the noisy observations from semantic segmentation models, is to leverage prior knowledge and learn the object's trajectory within the scene.

This project aims to overcome limitations in visual semantic segmentation by utilizing a Multi Hypothesis Extended Kalman Filter algorithm to increase consistency across views [6] [18]. The Multi Hypothesis Extended Kalman Filter (EKF) is a popular method used to track various objects by simulating multiple hypothesis of the motion equations and linearizing the nonlinear state and measurement equations using Taylor series expansions.

The pipeline utilizes the SAM model to generate initial segmentation masks for a semantic query across video frames. Each segmentation mask serves as an initial estimate for the object's pose. The center coordinates of the bounding box enclosing these masks are utilized as the initial pose values

relative to the camera. By incorporating the initial segmentation mask, we establish an informed starting point for pose estimation, which is further refined over time using an Multi Hypothesis Extended Kalman Filter (MHEKF). This integrated produced tracking results that are both more accurate than the raw detection output and resilient to challenges such as occlusions and noisy measurements.

II. Related Works

A. Vision Transformer Models

A Vision Tranformer (ViT), is transformer model specifically trained for visual recognition [10]. Unlike a Convolution Neural Network (CNN), ViT, utilize the attention mechanism of transformers to capture semantic relevance within images allowing it to attend to different input image regions, dynamically focusing on important visual features and their interactions [10]. The attention mechanism of ViT has been extended to language tasks, enabling models to develop a semantic understanding of the visual content they observe. Notable examples of Vision-Language models that leverage this capability are CLIP and DINO [5] [12].

Contrastive Vision-Language Pre-Training (CLIP) model, which has "pioneered alignment between 2D visual and textual features by pre-training on large-scale image-text pairs" [5]. Vision Language Models, like CLIP have allowed us to easily recognize and classify objects within 2D images. The unique aspect of CLIP, is the fact that it is a "zero-shot" classification model with a robust open-vocabulary model, allowing for the classification and extraction of objects not previously shown to model. CLIP has been extended to a large collection of applications of computer vision tasks involving language abilities, object detection, image classification, image manipulation, image retrieval, etc. [5]. Pre-trained CLIP's semantic-image representations cannot be directly extended to spatial segmentation tasks without utilizing pixel level information. [7].

Similar to CLIP, Grounding DINO [13] is a vision-language model capable of zero-shot classification. Unlike CLIP, Grounding DINO is specifically trained to identify and localize objects based on semantic queries, accurately placing bounding boxes around them. While Grounding DINO excels at open object detection tasks, it should be noted that its capabilities do not extend to image segmentation [14].

B. Semantic Segmentation Models

There has been significant work to extend the open vocabulary classification of text to image to segmentation [5] [6] [8]. Open Vocabulary Semantic Segmentation is an open vocabulary segmentation model built on top of CLIP which can segment an image into categories. OVSeg, uses a finetuned version of CLIP trained on masked images and corresponding text descriptions. Furthermore, OVSeg uses mask prompt tuning, where blank areas in the mask ("zero token areas") are replaced with learnable prompt tokens. The mask prompt tuning method led to a significant improvement in OVSeg, giving it similar performance to popular specialist segmentation methods (Deeplab, SelfTrain, etc.) [5] without dataset adaptations. While OVSeg is extremely performative, the open vocabulary nature of segmentation gives way to a high level of ambiguity in predictions, leading to segmentations that are "false", leading to inconsistent predictions across perspectives of the same object.

Segment Anything Model (SAM), [13] is an image segmentation model, trained on the largest segmentation model, containing over 1 billion semantic masks. For a zero-shot method, SAM produces extremely high quality masks when compared to fully supervised models. SAM is designed to segment images using a variety of potential inputs, position, prompts, etc. During training, SAM, was prompted with CLIP image embedding, where the text embedding of the image from CLIP are given to SAM at inference time. The SAM model, benefiting from its larger size and training with CLIP, surpasses the OVSeg model by generating more precise and refined masks. Moreover, the SAM model's capabilities have been extended to enable consistent segmentation across frames in models like SEEM. SEEM achieves continuity between frames by utilizing learnable memory to retain mask history and prompts. With an initial mask and prompt, SEEM identifies visual similarities to the initial mask and prompts in subsequent frames to predict the masks for each frame. While SEEM attempts to do this, qualitatively its performance is quite poor. However, is unable to do multi-object tracking and lacks an identification system for objects. SAM-Track, a recent method that combines the Segment Anything Model with Grounding DINO, given each bounding box and corresponding semantic mask, an Associating Objects with Transformers (AOT) model is used to track the object [14] [15]. To match and segment multiple objects as efficiently as processing a single one, AOT employs an Identification (ID) mechanism to assign objects with unique identities and associate them in a shared high-dimensional embedding space [15]. AOT methods work fairly fast and in SAM-track the results are fairly proficient, however, still somewhat struggles with measurement segmentation noise.

C. State Estimation and Tracking

State estimation involves predicting object location using probabilistic laws and Bayesian filtering techniques [18]. The Extended Kalman Filter is a nonlinear state estimation tool that assumes that a Taylor Series linearization of the system functions as an adequate representation of nonlinearities. [18] Kalman Filters have been extensively used in Robotics and Vision problems as a method to perform tracking. Rather than relying on constant detection, Kalman Filters, are especially good a detection because they are able to track moving objects by making predictions using a dynamic model of the object. [17] [19]

D. Multi-Hypothesis Kalman Filter

Multiple Hypothesis Tracking (MHT), originally developed for visual tracking, builds a tree of potential track hypothesis

for each potential trajectory, allowing the algorithm to develop a collection of possible trajectories for each detection. [20] As more potential paths are considered, MHTs prune and remove trajectories that are heavily diverge from the global hypothesis [MHT]. Since its introduction in the 90s, MHTs have not been a widely used method for visual tracking [20]. However, the structure of MHT problems have been extended to the Kalman Filter giving filter variations like the Multi-Hypothesis Kalman Filter (MHKF) and the Multi-Hypothesis Extended Kalman Filter (MHEKF). A MHKF and MHEKF functions as Kalman Filters that can maintain multiple hypothesis for trajectories that form a probability distribution that sums to 1. Similar to MHT, MHEKFs and MHKFs prune trajectories by setting a probability threshold, and a maximum length of possible trajectories [21]. The performance of MHKF and MHEKF, is connected with its ability to prune branches quickly and reliably. Furthermore, if the pruning parameters are not well-tuned, it can lead to false positives trajectories, where incorrect hypotheses are considered valid tracks and more accurate tracks are prematurely discarded [21]. When dealing with multiple detections of varying confidences obtained from a segmentation model for a single query, applying the Multi-Hypothesis Kalman Filter (MHKF) directly would be insufficient without modifications in how new trajectories are created [20] [21].

METHODOLOGY

E. Dataset

This project utilizes the DAVIS data set, a Densely Annotated VIdeo Segmentation data set [10]. Due to the heavy computation cost for processing and applying SAM and Grounding DINO to perform segmentation on the high resolution videos from the data set, the data for this research was shortened in length to 3 seconds of 4K 60FPS video. For the video used for testing, it contained a fixed camera to simplify the dynamics and tracking of the segmented object.

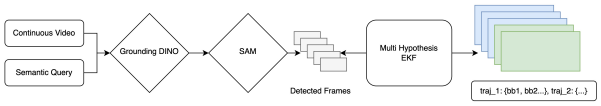

Fig. 1. Proposed Pipeline. Given the Continuous Video and Semantic Query, the pipeline passes the data into Grounding DINO and through SAM to return a colletion of frames with bounded boxes placed over guesses for the semantic query. Each frame represents a step within the trajectory observed in the video, the bounding box poses and the logits

F. Pipeline

Our pipeline initiates by processing a collection of frames from the [10] DAVIS dataset through SAM, which yields a corresponding collection of masks based on a specific query. Using these masks, Grounding DINO is employed to accurately place bounding boxes around the queried object. These masks and bounding boxes collectively serve as measurements for determining the object's pose. A given image/frame can return multiple separate masks for a given query, representing, guesses for where the semantic query is in the image. Each detection within a frame, returns a probability representing the confidence in the object as seen in 9. Every measurement within the process is accompanied with its own probability distribution, of whether the detection is accurate (1-p) the detection is a false positive. These parameters are kept track of until, it is passed through the MHEKF, which given the noisy measurements, creates a new guess of where the object is given its confidence score, the previous measurements, and the global hypothesis of the MHEKF. Given the filtered frames, they are recombined into a video, that can be observed for qualitative testing, in which the segmented object is tracked.

Fig. 2. Result of the SAM + Grounded DINO pipeline for the query "Yellow Skateboard." The green bounding box indicates a visually aligned object with a confidence score of 0.8.

G. Grounding DINO

The grounding DINO technique is employed to establish a similarity probability distribution between embedded regions within an image and embedded semantic classes. By computing the cosine similarity score (1) between the embedded image and the semantic class, we can assess the visual alignment. While DINO may not achieve 100% accuracy, it demonstrates a zero-shot accuracy of 60.7% on the COCO data set, making it a semi-reliable but noisy visual detector for open classes [13]

$$S_C(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} \tag{1}$$

H. Segment Anything Model with Grounding DINO

To perform semantic segmentation in this experiment, we utilize the Segment Anything Model (SAM) in combination with Grouping DINO. This model takes two inputs: a semantic

class and an RGB image. It then generates outputs including bounding boxes, segmentation masks, and confidence scores. The pipeline begins with Grounding DINO, which provides the x, y location of the semantic input. This is followed by the application of the Segment Anything Model, which refines the segmentation process, resulting in more accurate object localization through improved bounding box and mask estimation. To narrow down the scope of the experiment, we restricted the number of classes used in segmentation to one. Additionally, only detections with confidence scores surpassing a 30% threshold were considered for further analysis. Figure 3 illustrates an output of the SAM + Grounding DINO pipeline, demonstrating the exact output. However, for more obscure or nuanced semantic categories such as "skateboard wheel," the confidence score is significantly lower, leading to missed detections or incomplete segmentation. These challenges are attributed to the high levels of visual or semantic ambiguity inherent in such cases.

Fig. 3. Result of the SAM + Grounded DINO pipeline for the query "Skateboard Wheels" Despite, good performance on detection the majority of the semantic class model is unable to detect every single, wheel.

I. Dynamics

In our tracking system, we utilize a dynamic model to estimate the state of the object being tracked. The dynamic model assumes a straightforward position-velocity bounding box pose, where the state vector, denoted as μ, consists of the center coordinates (x, y) of the bounding box, as well as the width (w), height (h), and their respective velocities ($\dot{x}$, $\dot{y}$, $\dot{w}$, $\dot{h}$).

The state vector can be represented as follows:

$$\boldsymbol{\mu} = [x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]. \tag{2}$$

To model the dynamics of the system, we consider a linear setup where the next state (μ_{t+1}) can be calculated based on the current state (μ_t) and the time step (δt) using the following equation:

$$\mu_{t+1} = \mu_t + \delta t \cdot \mu_t \cdot F + Q, \tag{3}$$

where Q is the Gaussian process noise and F is the Jacobian of the dynamics given by:

$$F = \begin{bmatrix} I_4 & \delta t \cdot I_4 \\ 0_4 & I_4 \end{bmatrix} \tag{4}$$

J. Measurement Model

In our tracking system, we have position-only measurements available from our bounding box outputs. At each time step, the measurement can be represented as:

$$Y_{t+1} = C \cdot \mu_{t+1} + R \tag{5}$$

where Y_{t+1} is the measurement vector, μ_{t+1} is the state vector at time step $t+1$, C is the measurement Jacobin, and R represents the measurement noise.

The measurement Jacobin, represented as C, is given by:

$$C = \begin{bmatrix} I_4 \\ 0_4 \end{bmatrix} \tag{6}$$

This matrix selects the position components (x, y, w, h) from the state vector μ_{t+1} and maps them to the measurement vector Y_{t+1}, allowing us to obtain position measurements from the bounding box outputs.

The term R represents the measurement noise, accounting for uncertainties in the noisy detection measurements.

K. Multi-Hypothesis Extended Kalman Filter

1) Extended Kalman Filter: Our MHEKF, utilizes a standard Extended Kalman Filter algorithm to compute individual trajectories. The prediction and update steps for each hypothetical trajectories are the same. In the prediction step, the EKF generates the predicted state estimate $\bar{\mu}_{t+1}$ and the corresponding covariance matrix $\bar{\Sigma}_{t+1}$ based on the previous state estimate, $\bar{\Sigma}_t$.

In the update step, the EKF incorporates the measurement Y_t from our SAM (Segment Anything Model) and Grounding Dino Detector. It computes the Kalman gain K_t. The updated state estimate μ_{t+1} is then obtained by combining the predicted state $\bar{\mu}_t$ with the measurement innovation.

Prediction Step:

$$\begin{aligned} \bar{\mu}_{t+1} &= F * (\mu_t) \\ \bar{\Sigma}_{t+1} &= F_t \Sigma_t F_t^T + Q_t \end{aligned} \tag{7}$$

Update Step:

$$K_t = \bar{\Sigma}_t H_t^T (C_t \bar{\Sigma}_{t+1} C_t^T + R_t)^{-1} \tag{8}$$

$$\mu_{t+1} = \bar{\mu}_t + K_t(Y_t - h(\bar{\mu}_{t+1})) \tag{9}$$

$$\Sigma_{t+1} = (I - K_t C_t)\bar{\Sigma}_t \tag{10}$$

2) Multi-Hypothesis EKF: Our modified MHEKF, follows the generalized formula where for each previous trajectory and each detection seen at that time step, we create a new trajectory with weight α and measurement constant γ. The measurement weight γ is directly derived from the confidence score of the bounding box. The MHEKF process begins with setting the initial state to the various detections at $t = 0$. We also define a collection of state estimates $\boldsymbol{\mu}$ and covariance estimates $\boldsymbol{\Sigma}$ from $k = 1$ to the number of trajectories, and weights $\boldsymbol{\alpha}$ from

$k = 1$ to LN. The weight α is initialized by normalizing the confidence scores into a probability distribution.

The MHEKF algorithm performs an Extended Kalman Filter (EKF) on all combinations of initial trajectories and initial measurements. Given these trajectories, we use equation 11 to calculate the trajectory's current weight, $\alpha_{ik_t|t}$ for each trajectory k and detection i at time t, considering the measurement weight γ_i, the previous weight $\alpha^k_{t|t-1}$, and the probability of the measurement given the previous trajectory $p^{(ik)}(y_t|y_{1:t-1})$. The sum in the denominator ensures that the weights are properly normalized, creating a probability distribution that sums to 1.

$$\alpha_{ik_t|t} = \frac{\gamma_i \cdot \alpha^k_{t|t-1} \cdot p^{(ik)}(y_t|y_{1:t-1})}{\sum_{i=1}^{M}\sum_{k=1}^{LN} \alpha_{ik_t|t} = \gamma_i \cdot \alpha^k_{t|t-1} \cdot p^{(ik)}(y_t|y_{1:t-1})} \tag{11}$$

Typically a Multi-Hypothesis Kalman Filter is used where there are multiple Dynamics Models one hopes to follow, or in situations where there are multiple objects, however, in this context our data returns multiple guesses at a given time step, with different confidences that could be incorrect or correct. For this reason at every individual detection at a given time step, two trajectories are stored. The first trajectory represents a scenario where the detection is a false positive. In this case, we set $\gamma_{\text{false}} = 1 - \gamma$ and $\gamma_{\text{true}} = \gamma$. We use the EKF predict step without performing the update step to estimate the state and covariance. This allows us to track the trajectory assuming that the detection is incorrect. By appending the two tracks, we have one trajectory where $\alpha = \gamma_{true} * (...)$, representing the false positive scenario, and another trajectory where $\alpha = \gamma_{false} * (...)$, representing the true positive scenario. This approach allows us to consider both possibilities and provides flexibility in the tracking process. Following tracks forming we prune the branches that end up with low α_k weight through thresholding $\alpha_{threshold}$, and setting a maximum number of hypothesis, c_h . Finally, we re-normalize α_k again to add to 1.

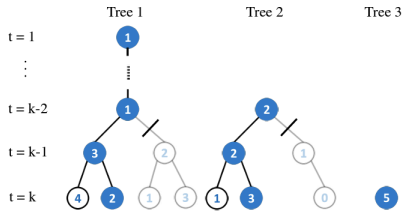

Fig. 4. N-Scan Pruning, pruning process for MHEKF depicted where different branches are removed either due to hitting the maximum length or falling below the weight threshold [21].

III. Results and Discussion

For this experiment, qualitative results were primarily used to observe the performance of the Multi-Hypothesis Extended Kalman Filter (MHEKF). Due to the uniqueness of the semantic queries used in our experiment, it was challenging to establish a baseline or find a suitable training set that included segmented or tracked instances of these unique queries. As a result, our analysis is limited to assessing the performance of the MHEKF within the context of the quality of the bounding box.

Fig. 5. Raw vs Filtered pose of trajectories semantic input "converse", max trajectories = 11. Top, unfiltered raw measurements for 11 frames. Bottom filtered pose estimation for 11 frames.

1) Visual Qualitative Tracking: For the primary video used during experimentation, "skate-is" from the 2017 DAViS set, we tested out tracking of three different open vocabulary semantic inputs, that were found in the scene: "converse", "person", "black hat". We compared the raw data results, where each detection was plotted as a pose measurement, to the filtered results obtained using the Multi-Hypothesis Extended Kalman Filter (MHEKF). The final video frame displayed with the bounding box trajectory of the results for raw and filtered approach is shown. For the more unique semantic queries like "converse" or "black hat" the average confidence score had low variance however, is on average extremely lower hovering around 0.52 ("converse), and 0.32 ("black hat") in comparison to "persons" 0.83. Hence, for the unique semantic queries so at each time step, the possible detections were generally equally probable.

In all query cases, as we increased the threshold for the maximum number of trajectories, we observed better tracking results, that closely followed the object. The difference in performance is observed in Fig. 7. This outcome aligns with expectations, as a higher maximum number of trajectories

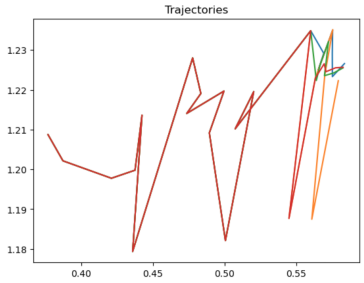

Fig. 6. MHEKF pose of trajectories semantic input ”converse”, max trajectories = 15.

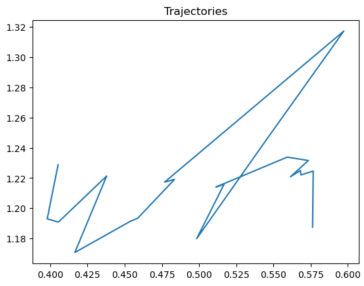

Fig. 7. MHEKF pose of trajectories semantinc input ”converse”, max trajectories = 8.

allows for a larger range of potential paths to be generated at each time step. As the MHEKF prunes possible trajectories, it is crucial to avoid premature pruning, which can lead to the removal of valid solutions or selecting a false positive confidence score as the final trajectory. Allowing for a larger number of possible paths at each time step enables the global hypothesis to converge towards the more expected state as more measurements are recorded.

Furthermore, the successful handling of occlusions in the filtered output, as demonstrated in Fig. 8, highlights the effectiveness of the dynamics model and filtering employed in the MHEKF algorithm. This robustness is particularly notable when the object is temporarily absent from the scene, as the MHEKF's prediction-only step compensates for the lack of measurements. This suggests the potential for further improvement by incorporating a visual transformer that compares the predicted state with previous measurements, allowing for more accurate pose estimations in scenarios with limited measurements.

This analysis highlights the importance of appropriately

Fig. 8. The query ”human” shows a better estimation with a maximum trajectory count of 100 (left) compared to 1 (right). Increasing the maximum number of hypotheses improves the overall estimation.

Fig. 9. Two Time Steps Shown of the MHEKF tracking semantinc input ”converse”, max trajectory 15, in the seen. During the time step where the skateboarder is behind the table his shoes are not detected and we receive no measurement for its location. The MHEKF estimated output captures the general position of the shoe while the raw measurement is unable to.

setting the maximum number of trajectories in the MHEKF to strike a balance between capturing diverse possibilities and avoiding premature pruning. It contributes to optimizing the performance of the tracking system and ensuring more accurate and reliable results.

IV. Conclusion

When combined with learned approaches such as semantic segmentation, MHEKF can do a noticeable job at providing higher quality continuous tracking/detection of a given object, than the raw data output. The qualitative results indicate the effectiveness of the MHEKF in handling uncertainties and variations in object appearances. However, to comprehensively assess the performance of the model and the visual detection algorithm, it is essential to compare the results against state-of-the-art models like YOLO, particularly on their defined categories.

Furthermore, leveraging visual learning techniques found in SEEM and YOLO could enhance the MHEKF by assigning unique feature IDs to detections, reducing noise and enhancing the discriminative power of the tracking system.

In summary, the MHEKF, in combination with semantic segmentation and potential integration of visual learning techniques, presents a promising approach for achieving robust and accurate object tracking.

V. Acknowledgements

I would like to thank Dr. Mac Schwager for his guidance and many clever ideas for this project. I also want to thank Olaoluwa Shorinwa and Jun En Low for their help throughout this quarter and for their advice on the project.

REFERENCES

[1] Mildenhall, Ben, et al. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. arXiv, 2020

[2] X. Zou et al., 'Segment Everything Everywhere All at Once', arXiv [cs.CV]. 2023.

[3] Fan, Zhiwen, et al. NeRF-SOS: Any-View Self-Supervised Object Segmentation on Complex Scenes. arXiv, 2022

[4] Vora, Suhani, et al., NeSF: Neural Semantic Fields for Generalizable Semantic Segmentation of 3D Scenes. arxiv, 2021

[5] A. Kirillov et al., 'Segment Anything', arXiv [cs.CV]. 2023.

[6] Kim T, Park T-H. Extended Kalman Filter (EKF) Design for Vehicle Position Tracking Using Reliability Function of Radar and Lidar. Sensors. 2020; 20(15):4126.

[7] Ha, Huy, and Shuran Song. 'Semantic Abstraction: OpenWorld 3D Scene Understanding from 2D Vision-Language Models'. ArXiv [Cs.CV], 2022,

[8] Lüddecke, Timo, and Alexander S. Ecker. 'Image Segmentation Using Text and Image Prompts'. ArXiv [Cs.CV], 2022, http://arxiv.org/abs/2112.10003. arXiv.

[9] Barreiros, M.d., Dantas, D.d., Silva, L.C.d. et al. Zebrafish tracking using YOLOv2 and Kalman filter. Sci Rep 11, 3219 (2021).

[10] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, 'The 2017 DAVIS Challenge on Video Object Segmentation', arXiv [cs.CV]. 2018

[11] A. Dosovitskiy et al., 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale', CoRR, vol. abs/2010.11929, 2020.

[12] M. Caron et al., 'Emerging Properties in Self-Supervised Vision Transformers', arXiv [cs.CV]. 2021.

[13] S. Liu et al., 'Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection', arXiv [cs.CV]. 2023.

[14] Y. Cheng et al., 'Segment and Track Anything', arXiv [cs.CV]. 2023.

[15] Z. Yang, Y. Wei, and Y. Yang, 'Associating Objects with Transformers for Video Object Segmentation', arXiv [cs.CV]. 2021.

[16] Z. Yang, J. Miao, X. Wang, Y. Wei, and Y. Yang, 'Scalable Multi-object Identification for Video Object Segmentation', arXiv [cs.CV]. 2022.

[17] M. de F. Coelho, K. Bousson, and K. Ahmed, "An Improved Extended Kalman Filter for Radar Tracking of Satellite Trajectories," Designs, vol. 5, no. 3, p. 54, Aug. 2021, doi: 10.3390/designs5030054.

[18] S. Y. Chen, "Kalman Filter for Robot Vision: A Survey," in IEEE Transactions on Industrial Electronics, vol. 59, no. 11, pp. 4409-4420, Nov. 2012, doi: 10.1109/TIE.2011.2162714.

[19] K. Saho, 'Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design', Kalman Filters - Theory for Advanced Applications. InTech, Feb. 21, 2018. doi: 10.5772/intechopen.71731.

[20] C. Kim, F. Li, A. Ciptadi and J. M. Rehg, "Multiple Hypothesis Tracking Revisited," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 4696-4704, doi: 10.1109/ICCV.2015.533.

[21] M. Quinlan and R. H. Middleton, 'Multiple Model Kalman Filters: A Localization Technique for RoboCup Soccer', 06 2009, vol. 5949, pp. 276–287.