Multi-Object Tracking, Segmentation and Validation in Thermal Images

Mircea Paul Muresan
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Mircea.Muresan@cs.utcluj.ro

Radu Danescu
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Radu.Danescu@cs.utcluj.ro

Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
Sergiu.Nedevschi@cs.utcluj.ro

***Abstract*—In this paper we present a novel multi-object tracking and segmentation approach that works on thermal images and is able to track objects at bounding-box and instance mask levels. Furthermore, we present a novel object validation module, which is necessary because only specific classes of objects are tracked and classifiers and detectors can be subjected to errors such as misclassifications, false detections, erroneous instance masks and missed detections. One of the key difficulties in multi-target tracking is the unknown correspondences between measurements and targets also known as the data association problem. To address this issue the proposed object tracker uses a feature engineering data association approach that exploits multiple features which include structure, appearance, size, context, and motion in the region given by the instance mask. Moreover, an original strategy has been designed for dealing with motion uncertainty, based on optical flow and multiple motion models to better predict the future position of objects in the scene. The proposed method runs in real-time and has been evaluated on an international thermal tracking benchmark showing competitive results.**

Keywords—MOTS, data association, motion models, semantic segmentation, instance segmentation, thermal imaging

I. INTRODUCTION

Multi-object tracking and segmentation (MOTS) [1] is a challenging problem in computer vision, and has a wide range of applications in fields such as surveillance, advanced driving assistance systems, autonomous driving, and robotics. The goal of multi-object tracking (MOT) [2] is to accurately reidentify objects bounding boxes in successive frames and estimate and filter their trajectories, while object segmentation aims to identify and separate individual objects within the image and provide a semantic class for each object. MOTS builds upon the tracking task by using instance segmentation masks[3] instead of bounding boxes when creating the object trajectories thus extending the precision to the pixel level. Furthermore, the instance segmentation module can provide more precise information about the position of objects in the scene even in occluded situations, and the semantic segmentation [4] module can provide more context information which can aid the data-association process from the tracking component. The segmentation and tracking tasks are intertwined in real-world applications [5] and by exploiting their synergies the tracking performance and the overall decision-making process of a self-driving car can be improved. The most dominant multi-object tracking paradigm among state-of-the-art MOT algorithms that were used in recent years is tracking-by-detection [2][6][7][8][9]. This MOT paradigm consists of two steps. In the first step object detections from each frame are extracted and then, in the second step, detections are linked to form the object trajectories and maintain object identities across frames.

One of the most important stages in the multiple object tracking pipeline is by far the data association [10]. The poor handling of the data association step can lead to bad tracking results which can have disastrous effects. The issues that can affect the performance of the data association can be split into two main categories: origin and motion uncertainties. The origin uncertainty refers to the fact that there is no prior knowledge of how new sensor data relates to previous measurements. On the other hand, motion uncertainty refers to the fact that objects in the real world can exhibit multiple motion patterns, therefore a single motion model cannot accurately predict the position of all objects in the scene. In order to have a good tracking solution the data association issues have to be treated in a robust and efficient manner.

Thermal cameras have attracted a lot of interest in recent years in the automotive field[8][9][11], due to their ability to function during all seasons in day, or night scenarios and even in adverse weather conditions such as snow, rain, or foggy weather. Moreover, the usage of thermal cameras during the night improves the reaction time of human drivers due to the ability of the camera to detect living beings from large ranges. Furthermore, the thermal images do not saturate in the presence of lights from incoming vehicles, making thermal cameras a necessary enabling technology for the automotive field. The disadvantages of thermal images are that they usually have a lower resolution and do not contain as much information as color images making the data association step for a tracking application more difficult. For solving the above-mentioned issues, in this paper, we are proposing a multi-object tracking and panoptic segmentation solution that has been designed to work on thermal images. To the best of our knowledge, this is the first work in the literature that tackles the problem of multi-object tracking and segmentation on thermal images. In summary, this paper brings the following contributions:

- We propose an original pipeline that combines instance segmentation, semantic segmentation, and multi-object tracking on thermal images for the task of multi-object tracking and segmentation. The proposed framework can output both tracked bounding boxes and masks depending on an option from the user.
- Furthermore, we propose a novel detection validation scheme because object detectors are imperfect and may provide erroneous results. Moreover, an original mask refinement strategy is introduced for refining the obtained instance mask results.
- We present a novel feature engineering data association approach for dealing with the data association issue, and an original optical flow-based multi-motion model selector approach designed to work for multiple objects that can have diverse motion patterns.

The proposed solution has been extensively evaluated and a link to a short movie illustrating the results can be found at:

https://youtu.be/Bit3DkKeyK4 The rest of the paper is organized as follows: Section 2 provides a review of the state of the art in the field of multi-object tracking. Section 3 describes the proposed approach in detail. Section 4 presents experimental results and a performance evaluation of our approach. Finally, in Section 5 we conclude the paper and discuss future work.

II. Related Work

A. Multi-Object Tracking

The most common sources of information when computing the similarity between detections in consecutive frames, in a tracking-by-detection framework, are object appearance and motion. There are three directions in the literature in which the features are extracted for the data association task, each with its advantages and disadvantages: feature engineering approaches (or model-based), data-driven methods, and a combination between feature engineering and data-driven.

The authors in [2] present a feature-engineered cost function for computing the similarity between tracks and detections. This function incorporates various features such as object dimension and color histograms. The motion similarity is fused with the appearance score and contains the L2 norm computed between the position of the track and detection. The Hungarian algorithm [12] was used to find the best track and detection mappings. In [13] Brehar et. al. presented a tracking approach that also combines feature-engineered motion and appearance cues. The appearance cost was composed of a weighted combination of multiple scores including IoU, differences in object dimensions, and the uniform LBP of the region of interest. The motion score contained information from optical flow and differences in position between track and detection. The work of Yu et. al. [14] presents another feature engineering data association approach that uses edge orientations transferred to the Fourier domain to obtain a very fast object-tracking solution in the thermal domain.

Data-driven approaches compute the similarity score between detections and tracks using methods that involve learning. In the work presented in [6] the authors trained a Siamese network to distinguish between two objects in the thermal domain using the optical flow data. To obtain more rich information that could be used in the data association function, the Siamese network contained multiple convolutional layers and fused the data from shallow and deep layers. In [7] the authors combined a spatial transformer network with a multi-stage region proposal network (RPN). Furthermore, they fuse features from shallow and deep layers in the Siamese network which contain both spatial and semantic data to obtain a more compact feature representation. The multi-stage deep feature fusion network was used for tracking objects in the thermal domain.

Some solutions from the literature combine feature-based approaches with data-driven methods to obtain more robust trackers, at the cost of increased complexity. For example, in [9] the authors create a weighted function using multiple engineered descriptors that capture the texture, structure, and size of objects as well as a Siamese neural network. Another solution presented in [15] combines engineered features with data-driven results using gradient boosting. Features such as dimension change, and position change, are combined with the results of Siamese CNN that takes as input the pixel values in normalized LUV color format as well as optical flow components.

B. Multi-Object Tracking and Segmentation

MOTS has been introduced as an extension of MOT to improve the data association step by incorporating richer visual cues and overcoming some of the limitations given by using bounding boxes. The first MOTS method that was applied to color images was introduced by Voigtlaender et al. [1]. The authors created a baseline method called TrackRCNN which builds upon MaskRCNN[3] by integrating 3D convolutions to treat the temporal information. In [16] the authors propose a method that exploits instance segmentation and bounding box detection. This approach, entitled MOTSFusion generates segmentation masks for each bounding box and then uses 2D optical flow to generate short tacklets. The tracklets are then fused to obtain the precise reconstructed 3D object motion, which in turn is used to recover occluded objects. Yan et. al. in [17] use depth and optical flow information together with box labels to generate more accurate instance labels. The objects are associated using a similarity function that uses the position of the items in the scene. The authors use bidirectional greedy matching (from the current to the previous frame and vice versa) for optimal assignment instead of the Hungarian method and a Kalman Filter to predict the position of objects in future frames. In the ReMOTS solution[18], the authors first use a neural network to detect objects in each frame of a video stream and then use a combination of motion-based and appearance-based features to perform the data association for the objects from the current frame with the ones from the previous frame. The objects are tracked using a combination of two tracking algorithms (SORT[19] and DeepSORT[20]), the tracking approach is selected based on the scenario and quality of the detections. The instance segmentation network used in this approach is MaskRCNN[3].

III. Proposed Solution

A. Pipeline and Validation Scheme

In this section, we will present a framework capable of performing panoptic segmentation (semantic and instance segmentation) and tracking on bounding boxes and instance masks for each object of interest. The application runs on images from the thermal domain and uses an additional validation step for verifying the correctness of the detections and object classes for the classes of interest. The validation scheme is useful because the classification step may be erroneous and object detectors may be subjected to errors such as missed detections or false detections. Furthermore, we are not tracking all objects from the scene only objects that have specific semantic classes. Therefore, having a validation module for the object classes and detections is very important in identifying the objects of interest. The application uses three different data-driven approaches for the validation tasks and a model-based approach for the data association and tracking part. The results from the data-driven and model-based approaches are fused to obtain the final results at the mask and box level. A diagram showing an intuitive depiction of the proposed pipeline is shown in Figure 1. It is worth noting that the focus of the paper is not on the object detectors and the used object detectors can be replaced in the proposed pipeline with other models from the literature. In the proposed approach semantic segmentation has been performed using an ERFNet[21]. This model runs in 8ms on the GPU on input images having a resolution of 640x480 and has been trained to identify 25 semantic classes.

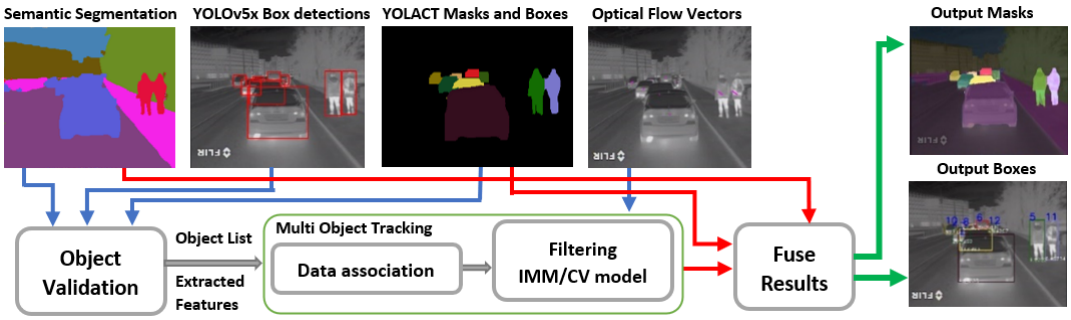

Figure 1. The proposed multi-object tracking and segmentation for thermal images pipeline

A combination of two object detectors, YOLOv5[22] and YOLACT [23], has been used for detecting instances. The reason for using two models has to do with the validation step. The validation aims to reduce the errors caused by object detectors or classifiers. To validate the classification and object detection results, at least three distinct algorithms need to analyze the same object instances. In this paper, we have used the output of YOLACT, YOLOv5, and the semantic segmentation obtained with ERFNet. The object detector used is a YOLOv5x which was trained on the FLIR ADAS dataset. The training set was modified such that only the following classes of interest are detected: car, bus, truck, person, cyclist, motorcycle, bicycle, traffic sign, and semaphore.

One issue of the YOLACT model is that if the detection threshold is larger than 50% the detector does not capture all objects of interest from the scene. On the other hand, if the threshold is below 50% there may be many erroneous detections. To overcome this issue, we have set the confidence threshold of the YOLACT detection low at 10% such that many detections are generated, and then the intersection over union (IoU) is computed between the YOLOv5 detections and the ones generated by YOLACT. For each YOLOv5 object, two YOLACT objects that have the largest values for IoU are stored. For quick access, the indices of the objects from the detectors are stored in a lookup table, with the mention that the IoU value of the associated items must not be below a threshold, set in our case to 0.8.

In the validation step, we verify if the semantic classes of the YOLOv5 correspond to the class of at least one of the two YOLACT objects. First, the class of the object with the largest IoU is tested then the other object's class is tested. In the case in which the classes correspond, it means that the object being tested is valid. If both classes are valid, the YOLACT detection having the largest IoU is considered for future steps. Otherwise, if the semantic class has not been validated, we test the object class of the YOLOv5 detection with the dominant semantic class obtained from the semantic segmentation image in the region of interest (ROI) given by the YOLOv5 detection. So, an algorithm is implemented, using the CUDA framework, that generates the histogram in the ROI given by the YOLOv5 detection in the semantic segmentation image and chooses the semantic class corresponding to the bin that has the most of the most votes in the histogram. In case two of the semantic classes from the three algorithms match we say that the bounding box has a valid semantic class. When all three classes are different the semantic class of the object is unknown.

In the last step of the algorithm, we verify if there still are YOLACT objects that have a confidence score over 50% and have not been previously validated using the YOLOv5 detections. In case there are such objects, their semantic class is compared with the semantic class coming from the semantic segmentation image taking as ROI the YOLACT detections. In a similar manner, we also verify if there are any YOLOv5 objects which have not been previously associated, and their semantic class is verified using the semantic segmentation image. At the end of the algorithm, we obtain a list of bounding boxes for which we have the validated semantic classes. Furthermore, the instance masks can be obtained from the YOLACT associations. In case there is no instance mask for an object, the mask is generated using the semantic segmentation image taking into account the region of interest given by the detection bounding box and the object semantic class. It is worth mentioning that for obtaining a good running time all algorithms have been parallelized on the CPU and GPU.

B. Mask Refinement

The instance masks can, at times, be imperfect containing different artifacts or they can be incomplete not covering entirely the object of interest. For solving these issues, a fast refinement approach has been implemented.

The used object detector produces for each object instance a mask image having the same dimension as the original image. The produced images are binary and contain a value of 1 where the object mask is localized and 0 in the rest of the image. Since the bounding box of the object of interest is known, the first step is to apply an opening morphological operation using a 7x7 kernel such that artifacts that are produced and glued to the instance masks of the objects of interest are split. Then we perform a two-step labeling algorithm [24] using classes of equivalence to determine the object clusters from the region of interest given by the object bounding box and we also compute the object area of each cluster in this process. We have chosen a two-step labeling approach with classes of equivalence due to its superior speed in contrast to other methods. Finally, we select as the object mask the cluster from the ROI which has the maximum area.

After applying the procedure mentioned above, we extend the object masks using the semantic segmentation image considering the dominant semantic class. A new pixel is added to the instance mask of an object only when it has a class equal to the dominant semantic class inside the region of interest and it neighbors an instance mask pixel. The mask is extended only within the ROI given by the bounding box detection. Each object instance will have associated to it a

mask image. After the tracking algorithm, all masks will be fused in one image using a color code associated to the unique id of the tracked item. It is worth noting that before fusing the masks, the objects will be ordered in ascending order based on the distance of the object to the camera. This distance is obtained using the algorithm presented in [13]. In Figure 2 it can be observed that the object detector has generated an erroneous instance mask and multiple artifacts for the object with pink. By using the proposed refinement approach, the unwanted artifacts are removed, and the mask is corrected with the help of the semantic segmentation image.

By annotating more images some erroneous situations can be avoided, however, it is impossible to cover all scenarios that may appear in the driving environment and a refinement procedure ensures more robust results even in difficult situations. Moreover, it is not common for erroneous results to appear however if they appear it is important to have a strategy to mitigate the effects of a bad instance mask.

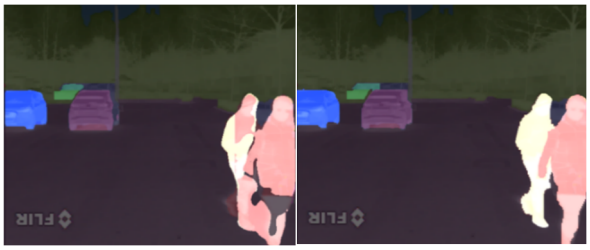

Figure 2. In the left-hand side the erroneous instance mask. In the right-hand side the corrected instance masks are displayed

C. Data Association

The tracking solution presented in this paper follows a tracking-by-detection framework, where the cost function used for associating tracks and detections uses an appearance and a motion score. The proposed tracking approach has been designed to consume few resources having the possibility to run on embedded devices. The input of the tracking module consists of a set of detections which include a set of features extracted and the instance masks for each object which will be used in the data association process. The output of the tracking module is a set of tracks that have a unique id and a filtered trajectory. The output tracks are given at both bounding box and mask levels. The components of the proposed method include clutter removal, similarity cost computation, data association between track and detection, tracking update, and refinement. A validation gate around the predicted position was used for reducing the number of associations and in consequence improving the running time. Only the detections that fall within the validation gate of a track are considered in the data association process.

In this paper, we have implemented a feature-engineered data association approach for computing the similarity score between tracks and detections. We build upon the state of the art by creating a similarity score that is computed as a weighted function having multiple terms. By using a model-based approach for data association, the running time of the solution is improved and we are able to see the contribution of each feature used and come up with a solution in case some of the results are not as expected.

The appearance score offers the tracker the possibility to distinguish between objects using visual features. Moreover, the appearance of the tracks is able to adapt based on the changes that can appear to objects due to different illumination, deformations, or point of view changes.

The appearance score between track i and detection j contains several visual features and the equation of the appearance score can be seen in (1). The value of $\alpha(i,j)$ is minimal for the same object instance.

$$\alpha(i,j) = \frac{w_1}{\vartheta(i,j)} + w_2\gamma(i,j) + w_3\tau(i,j) + w_4\rho(i,j) + w_5\mu(i,j) + w_6\sigma(i,j) + w_7\delta(i,j)\omega(i,j) + (1 - miou(i,j))w_8 + w_9\theta \quad (1)$$

The meaning of the parameters is the following: ϑ represents a part-based orientation cost, γ is a semantic segmentation cost, τ is the overlapping cost, ρ is the uniform local binary pattern (ULBP) cost, μ and σ are the mean and variance costs, θ is a dimension cost, δ and ω represent the classification probability and $miou$ is the mask intersection over the union. The values of the weights have been determined experimentally. The scores $\gamma, \tau, \mu, \sigma, and\ \theta$ are computed by using the L2 norm between the value contained in the track and the one contained in the detection, while miou is the mask intersection over union computed between the instance masks of the track and detection. Features such as ULBP, object mean and variance are computed similarly to the approach presented in [13] with the mention that the cost is computed only in the region given by the instance segmentation mask, not the whole bounding box.

The score that includes the classification probability is computed using the L2 norm between the classification probability stored in the track (of the previous detection) and one of the current detections. The value of ω is 1 when the semantic classes of the track and detection match and has a high value otherwise (in our case the value is 2000).

For incorporating context information, we include a histogram of the semantic classes in the ROI given by the object bounding box. We reason that the context information for the same object instance does not change drastically between frames, and the difference between the histogram values of the track and detection is minimal for the same object. The semantic cost between a track and detection is computed as in equation (2), where $\gamma(x)$ is the value from the histogram of the bin x and 25 is the number of classes used.

$$\gamma(i,j) = \sum_{k=0}^{25} |\gamma_i(k) - \gamma_j(k)| \quad (2)$$

Even though thermal radiation does not depend on any external light source, the combination of the pedestrian clothes or even some materials and the thermal radiation leads to unique textural and structural patterns that can be exploited by data association functions. Furthermore, even though the environment in which the thermal images are captured plays an important role in the apparent temperature of the target, due to the fact that the framerate of the camera is sufficiently high and the characteristics of each track are updated at each frame the multi-object tracking performance is not affected. We observe that object texture does not change very much between frames. It is therefore a good feature to use when measuring the correlation between tracks and detections. We aimed to capture the texture of each object, even in partial occlusion situations, by combining a grid-based matching technique with edge orientations, using multiple histograms matching between the i^{th} track and j^{th} detection. In the first step, we compute the magnitude G and

orientation θ of the gradient as presented in (3) using the image derivatives I_x *and* I_y, which were obtained using a Sobel descriptor.

$$|G| = \sqrt{I_x^2 + I_y^2};\ \theta = \arctan\left(\frac{I_y}{I_x}\right) \quad (3)$$

We then divide the obtained orientation by a factor of 18 to have at most 20 orientations values. The bounding box corresponding to the object of interest is then split into a 3x3 grid and for each cell of the grid, we compute an orientation histogram that has 20 bins. Each pixel that is in a cell, casts a vote in the histogram that corresponds to that grid cell provided that in the instance mask of the object in that specific position there is a value of 1, meaning there is a mask in that position. The voting is done using the gradient magnitude. The similarity for a grid cell is computed as shown in equation (4) and (5), where $\varphi(i,j)_x$ represents the similarity between a grid cell x of track i and detection j, ε is a threshold value which was determined experimentally and has the value 0.15, while $\vartheta(i,j)$ is the final expression for all grid cells. The value of $\vartheta(i,j)$ is minimal if the track and detection belong to the same object instance.

$$\varphi(i,j)_x = \begin{cases} 1, \left(\sum_{k=0}^{20} \frac{|H_i(k) - H_j(k)|}{20}\right) \leq \varepsilon \\ 0, otherwise \end{cases} \quad (4)$$

$$\vartheta(i,j) = \frac{w_1}{\sum_{x=0}^{8} \varphi(i,j)_x} \quad (5)$$

The motion score, $m(i,j)$, contains the difference in position, computed using the Euclidean distance, between the location of track i and the position of the detection j, with respect to the center position of the object expressed in 2D coordinates. The final similarity score between a track and a detection is composed by summing the appearance and motion scores. After computing the similarity scores between tracks and detections, an optimal assignment algorithm is used [12] for matching each track with its corresponding detection. After the matching has been done, each track is updated using the information from the detection and a filtering approach is employed to smooth the object trajectory.

D. Optical Flow based Model Selector

Objects in the real world can have different motion behaviors, therefore by using a single motion model we cannot adequately capture the position of road users. In this section we will present an original approach that combines multiple motion models using different strategies in order to accurately predict the position of the objects of interest.

The general idea of the proposed approach is to run two filters in parallel and select the states from the filter that is best suited for a certain object of interest by using information from the optical flow. The first filter is a Kalman Filter (KF) that uses a constant velocity motion model that has been designed to capture static objects and objects that are moving slowly while the second filter is an interacting motion model (IMM) [32] filter that combines constant velocity and a constant acceleration motion model. The interacting motion model filter has been designed to capture the dynamic motion of the maneuvering targets.

When designing the IMM filter a set of modes are first selected for describing the maneuvering of the target and there is an assumption that the object is always in one of these modes. It is worth mentioning that the running time increases with the number of modes. To obtain a real time performance in our solution we selected two modes, which can be described using a constant velocity (CV) and a constant acceleration (CA) motion model. Each mode has equal probability when a track is first initialized. Secondly, we model the transition between modes using a Markov chain that is represented using the transition probability matrix Π, that was identified experimentally (6).

$$\Pi = \begin{bmatrix} 0.97 & 0.03 \\ 0.06 & 0.94 \end{bmatrix} \quad (6)$$

Using the transition probability matrix, the new mode probabilities can be computed at each iteration using (7), where m is the number of modes, μ represents the mode probabilities, and $\bar{c}$ is the new mode probability.

$$\bar{c_j} = \sum_{i=0}^{m-1} \mu_i \Pi_{i,j}, j = \overline{0, m-1} \quad (7)$$

Since the state vector of the two used modes has different dimensionality (the CV model has 4 values in the state vector while the CA model has 6 values), we have extracted only the 4 common states from the two models and we operate with them. It is worth noting that even the update of the states and covariance matrix will be done to the values that involve the four elements (position on x and y and velocity on x and y dimensions). Then the mixing probabilities or mixing weights are computed as in (8).

$$w_{i,j} = \frac{\Pi_{i,j}\mu_i}{\bar{c_j}}, i,j = \overline{0, m-1} \quad (8)$$

After each KF from the model bank performs an update step obtaining a mean and covariance, a new mean and covariance will be computed for each filter using the mixing probabilities as a weighted sum of the means and covariances of each filter. The unlikely filter will receive a strong adjustment by the likely filter and the likely filter will receive only a small adjustment, process described in (9) (10).

$$\bar{x_j} = \sum_{i=0}^{m-1} w_{i,j} x_i \quad (9)$$

$$P_j = \sum_{i=0}^{m-1} w_{i,j} [(x_i - \bar{x_j})(x_i - \bar{x_j})^T + P_i] \quad (10)$$

The model probability at time stamp k is updated for each mode using the mode probability at the previous time stamp and the likelihood of each filter as shown in (11).

$$\mu_k^i = \frac{\mu_{k|k-1}^i L_k^i}{\sum_j \mu_{k|k-1}^j L_k^j}, i,j = \overline{0, m-1} \quad (11)$$

The final state estimation of our IMM uses a mixed estimate from each Kalman Filter as presented in (12) and (13), where $\overline{x_k^l}$ and $\overline{P_k^l}$ represent the predicted state and covariance for each filter from the filter bank at time stamp k.

$$\widetilde{x_k} = \sum_{i=0}^{m-1} \mu_k^i \overline{x_k^l} \quad (12)$$

$$P = \sum_{i=0}^{m-1} \mu_k^i [(\overline{x_k^l} - \widetilde{x_k})(\overline{x_k^l} - \widetilde{x_k})^T + \overline{P_k^l}] \quad (13)$$

The measurement covariance matrix used with the models from the model bank of the IMM filter have a different value depending on the object class. We are using 4 different measurement covariance matrices for the classes: car, pedestrian, rider and other. Additionally, to the IMM filter that uses two motion models, a KF that uses a constant velocity model is also included to the object tracker. The KF with the CV model uses only one measurement covariance matrix and has small values in the process noise covariance matrix. There are situations in which an IMM filer may perform poorer than a KF with a constant velocity model, and for this reason selecting the appropriate filter depending on the situation is necessary. Some of these situations include scenarios where objects are moving slowly (for example when the vehicle is in a parking lot), when the vehicle is moving on straight roads with respect to the ego vehicle inside the city or when a self-driving car is navigating in an environment with a lot of measurement noise such as in a heavy fog situation or with a malfunctioning sensor. For addressing these issues, a method of selecting the correct filter depending on the situation has been implemented by using the optical flow information.

Sparse optical flow may not provide reliable results from a qualitative point of view, while dense optical flow can be demanding form a computational side and may provide erroneous results on unstructured surfaces. Even though the individual trajectories obtained from the optical flow may be erroneous, by aggregating the results from multiple trajectories inside a region of interest we can obtain clues regarding the motion of objects in successive frames. The optical flow algorithm presented in [25] has been applied and several steps were performed for obtaining the length of the aggregated flow trajectory for an all objects of interest. To store the optical flow values 36 bins have been created, where, using the angle data, each optical flow vector casts a vote inside a bin. The mean values for the optical flow magnitude and angle are computed for each bin, after all flow vectors have voted inside the corresponding bins. Finally, the mean magnitude and angle corresponding to the bin where the majority of the votes were cast are selected as the main flow parameters for the region of interest. To find the distance corresponding to the identified flow parameters two points are needed. The first point $P1(x_1, y_1)$ is located in the center of the object of interest, and the second point $P2(x_2, y_2)$ can be computed using the point P1 together with the found length and angle as shown in (14) and (15).

$$x_2 = x_1 + length \times \cos\left(angle \times \frac{\pi}{180}\right) \quad (14)$$

$$y_2 = y_1 + length \times \sin\left(angle \times \frac{\pi}{180}\right) \quad (15)$$

The ratio between the Euclidean distance between the two points P1 and P2 and the object width gives us an indication regarding how much the object has moved between frames. If the ratio is sub unitary it shows that the object has not moved so much so the Kalman filter with constant velocity will be used, otherwise the IMM filter will be used. The tracked objects are sorted using the depth to the camera obtained using the algorithm presented in [3], and then all tracks (masks and bounding boxes) are projected on the image obtaining the final results as presented in Figures 3 and 4. A track management approach is used to handle objects that enter and exit the field of view, removal of the old tracks that have not been updated for a long time etc. A variety of track management approaches exist in the literature [2][9][13] and we will not elaborate on this aspect.

Figure 3. Multiple tracked pedestrians and vehicles inside a city.

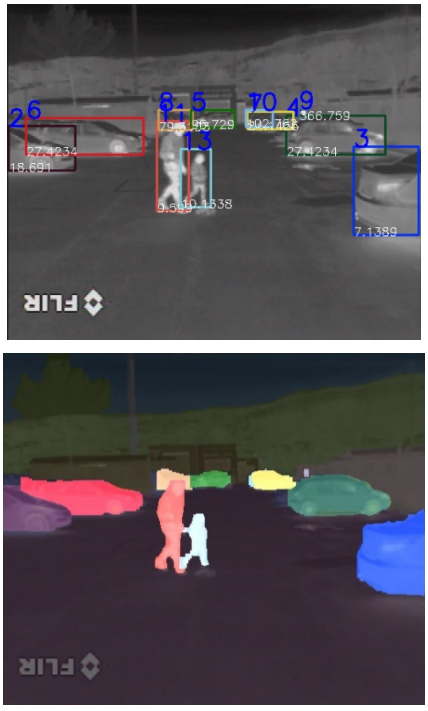

Figure 4. Tracked Bounding boxes and masks for objects inside a parking lot.

IV. Experimental Results

The proposed solution has been implemented in C++ and the used neural networks models have been trained in Pytorch and ported in Libtorch framework for obtaining C++ compatibility. OpenCV has been used for drawing, display and fusing the results in one image. CUDA and OpenMP

have been used for accelerating the speed of the proposed approach in order to obtain a real time performance. The solution was developed on an Intel Core i7-11370H processor having a 3.3 Ghz frequency and an Nvidia GForce RTX 3070 GPU.All datasets that were used for training the used neural network models from the proposed solution were split in three subsets 70% for training, 10% for cross-validation and 20% for testing. The training of the ERFNet network has been done on a thermal semantic segmentation dataset having more than 3500 images and 25 semantic classes. The following augmentation methods were used: Translations, Horizontal Flip, Gaussian Noise, Gaussian Blur and Gamma Contrast. The ERFNet was trained for 200 epochs, 100 for encoder and 100 for decoder using a learning rate of 5e-4 and a batch size of 4. The IoU obtained on the test set is 62.03%.

It is worth mentioning that a segmentation using FastSCNN[26] neural net was also tested, due to the high speed of the model, however it was not used due to the low IoU of 26.4%. YOLOv5 has been trained on the FLIR ADAS thermal dataset and fine-tuned on the CrosIR dataset [13] obtaining a 55.04% mAP. YOLACT uses a Resnet101-FPN and has been trained on the COCO dataset and then fine-tuned by training on 2000 thermal image detections that also had instance segmentations. The same augmentation techniques that were used when training the semantic segmentation model have also been used when training the YOLACT model. The instance and semantic segmentation datasets will be made publicly available in a future paper. The YOLACT with the proposed refinement approach was tested on 2000 thermal images annotated at instance level obtaining a mAP of 34.6% in contrast to the method that did not use the refinement that obtained a 31.2% mean average precision.

The tracking performance was tested on the PTB-TIR dataset [27] since there is no available dataset for tracking and segmentation in thermal images. However, since the proposed solution is also capable of outputting bounding boxes we are able to test the tracking performance on the PTB-TIR dataset. This dataset contains multiple sequences with thermal images each having manual annotations. The benchmark compares the performance of tracking approaches with respect to two metrics. The center location error (CLE) is one of the metrics which is described as being the average Euclidean distance between the ground truth position and the object position. A track is considered successful on the PTB-TIR benchmark, at a given frame, if the CLE is within a 20-pixel threshold. Another metric used on the PTB-TIR benchmark is the overlap score which measures the overlap ration between the bounding box area of the tracked object and ground truth. If a track has an overlap score that is above a threshold it is considered successful at a given frame. Different overlapping thresholds varying from 0 to 1 are used for ranking different tracking methods on the benchmark.

Table I. Results on the PTB-TIR dataset

Method	Tracking Success Score	Tracking Precision Score.
MDNet[28]	63.5%	79.3%
Proposed	**63.1%**	**82.6%**
DeepSTRCF[29]	62.7%	79.8%
VITAL [30]	62.2%	81.0%
TADT [31]	58.8%	73.4%
MLSSNet[8]	51.4%	70.6%

In Figure 5 the precision and success plots obtained on the PTB-TIR benchmark are presented.

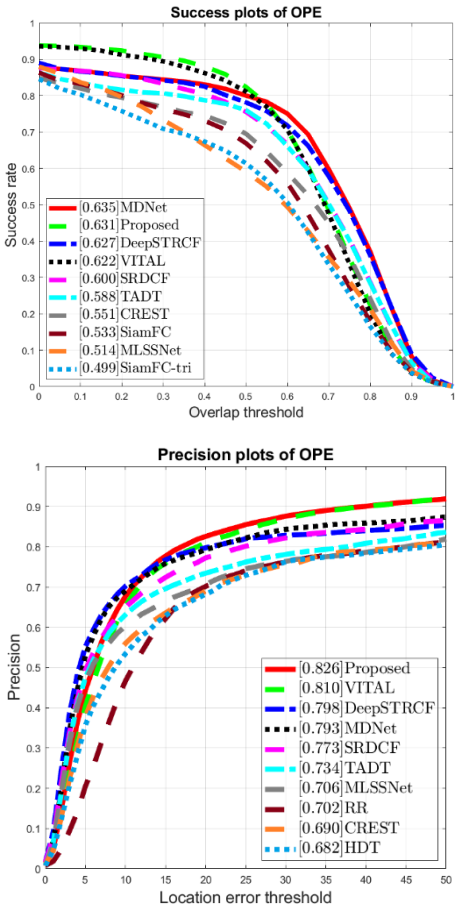

Figure 5. Graphical representations of the evaluation on the PTB-TIR dataset.

The evaluation has been done only on the sequences that were acquired from a vehicle mounted camera because the current solution is targeting intelligent vehicle applications. Moreover, only ten methods were displayed on the plots in order to keep the plots readable. For better readability the most relevant values from the two plots are also presented in Table I.

The prediction step of the proposed solution has also been tested using only the KF with CV motion model and the IMM filter with the CV/CA motion models obtaining a precision of 80.5% and 78.7% respectively. The association function remained the same for both of the cases mentioned above.

In contrast to some of the tracking methods presented in the PTB-TIR benchmark, the proposed solution is able to track multiple classes of objects not just pedestrians and is able to perform multi object tracking not just single object tracking. The running time of the tracking approach is 8 ms on the CPU. In Table II the running time of the proposed method on the intel core I7 on which it was developed and on a Nvidia Jetson Tegra TX2 are presented.

Table II. Running time of the solution on different platforms

Platform	Overall Average Running Time
Intel Core i7-11370H processor having a 3.3 Ghz frequency and an Nvidia GForce RTX 3070 GPU	100 ms
Jetson Tegra TX2	290 ms

V. Conclusions

In this paper, we have presented a novel multi-object tracking segmentation and validation pipeline that is able to track objects in thermal images at bounding box and instance mask levels. Object detections and their semantic classes are first validated using three data-driven approaches and are then passed to the multi-object tracker. Moreover, erroneous object instance masks are corrected using a novel refinement approach. The presented multi-object tracker uses an original model-based data association function that incorporates multiple terms in a weighted manner. The association function offers good results because it compares tracking features at the instance mask level not just at the bounding box level. Chiefly among the terms of the association function is an original grid-based multi-histogram of orientations matching approach that captures the thermal object's texture structure and a semantic segmentation histogram that captures context. Since objects in the real world can have multiple motion patterns, we presented a novel optical flow-based multi-motion model selector that can be used to choose the most suitable motion model for each object such that future positions are predicted more accurately. The proposed solution has been evaluated on the PTB-TIR benchmark and the running speed of the system has been tested on different computing platforms.

Acknowledgment

This work was supported by the Romanian Ministry of Education and Research, through CNCSUEFISCDI, project number PN-III-P4-ID-PCE2020-1700, within PNCDI III.

References

[1] P. Voigtlaender et al., "MOTS: Multi-Object Tracking and Segmentation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 7934-7943, doi: 10.1109/CVPR.2019.00813

[2] H. Karunasekera, H. Wang and H. Zhang, "Multiple Object Tracking With Attention to Appearance, Structure, Motion and Size," in IEEE Access, vol. 7, pp. 104423-104434, 2019, doi: 10.1109/ACCESS.2019.2932301.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, ''Mask R-CNN,'' in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2961–2969.

[4] D. Ando and S. Arai, "Semantic Segmentation Using HRNet with Deform-Conv for Feature Extraction Dependent on Object Shape," 2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS), Makasar, Indonesia, 2021, pp. 1-5, doi: 10.1109/ICORIS52787.2021.9649462.

[5] Y. -M. Song, Y. -C. Yoon, K. Yoon, H. Jang, N. Ha and M. Jeon, "Multi-Object Tracking and Segmentation With Embedding Mask-Based Affinity Fusion in Hierarchical Data Association," in IEEE Access, vol. 10, pp. 60643-60657, 2022, doi: 10.1109/ACCESS.2022.3171565.

[6] Q. Liu, D. Yuan and Z. He, "Thermal infrared object tracking via Siamese convolutional neural networks," 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Shenzhen, China, 2017, pp. 1-6, doi: 10.1109/SPAC.2017.8304241.

[7] X. Zhang, R. Chen, G. Liu, X. Li, S. Luo and X. Fan, "Thermal Infrared Tracking using Multi-stages Deep Features Fusion," 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 2020, pp. 1883-1888, doi: 10.1109/CCDC49329.2020.9164750.

[8] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan and H. Wang, "Learning Deep Multi-Level Similarity for Thermal Infrared Object Tracking," in IEEE Transactions on Multimedia, vol. 23, pp. 2114-2126, 2021, doi: 10.1109/TMM.2020.3008028.

[9] M. P. Muresan, S. Nedevschi, and R. Danescu, "Robust Data Association Using Fusion of Data-Driven and Engineered Features for Real-Time Pedestrian Tracking in Thermal Images," Sensors, vol. 21, no. 23, p. 8005, Nov. 2021, doi: 10.3390/s21238005

[10] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, Fundamentals of Object Tracking. Cambridge: Cambridge University Press, 2011.

[11] Chenglong Li, Wei Xia, Yan Yan, Bin Luo, and Jin Tang. Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation. IEEE Transactions on Neural Networks and Learning Systems, 2020.

[12] Kuhn, H.W. The Hungarian method for the assignment problem. Nav. Res. Logist. Q. 1955, 2, 83–97.

[13] R. D. Brehar, M. P. Muresan, T. Mariţa, C. -C. Vancea, M. Negru and S. Nedevschi, "Pedestrian Street-Cross Action Recognition in Monocular Far Infrared Sequences," in IEEE Access, vol. 9, pp. 74302-74324, 2021, doi: 10.1109/ACCESS.2021.3080822.

[14] Yu, X.; Yu, Q.; Shang, Y.; Zhang, H. Dense structural learning for infrared object tracking at 200+ Frames per Second. Pattern Recognit. Lett. 2017, 100, 152–159

[15] L. Leal-Taixé, C. Canton-Ferrer and K. Schindler, "Learning by Tracking: Siamese CNN for Robust Target Association," *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 418-425, doi: 10.1109/CVPRW.2016.59

[16] J. Luiten, T. Fischer, and B. Leibe, ''Track to reconstruct and reconstruct to track,'' IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 1803–1810, Apr. 2020

[17] L. Yan, Q. Wang, S. Ma, J. Wang and C. Yu, "Solve the Puzzle of Instance Segmentation in Videos: A Weakly Supervised Framework With Spatio-Temporal Collaboration," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 1, pp. 393-406, Jan. 2023, doi: 10.1109/TCSVT.2022.3202574.

[18] F. Yang et al., "ReMOTS: Self-supervised refining multi-object tracking and segmentation," 2020, arXiv:2007.03200.

[19] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple online and realtime tracking," *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, pp. 3464-3468, doi: 10.1109/ICIP.2016.7533003

[20] N. Wojke, A. Bewley and D. Paulus, "Simple online and realtime tracking with a deep association metric," *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 3645-3649, doi: 10.1109/ICIP.2017.8296962.

[21] E. Romera, J. M. Álvarez, L. M. Bergasa and R. Arroyo, "ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263-272, Jan. 2018, doi: 10.1109/TITS.2017.2750080

[22] G. Jocher; A. Chaurasia; A. Stoken; J. Borovec; NanoCode012; Y Kwon; M. Kalen; TaoXie; J Fang; imyhxy; Lorna; Y. Zeng; C. Wong; Abhiram V; D. Montes; Z Wang; C. Fati; J Nadar; Laughing; UnglvKitDe; V. Sonck; tkianai; yxNONG; P. Skalski; A. Hogan; D. Nair; M. Strobel; M. Jain; ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, Nov. 2022, https://zenodo.org/record/7347926

[23] Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-time Instance Segmentation. arXiv 2019, arXiv:1904.02689

[24] Umbaugh, S.E. (1997). Computer Vision and Image Processing: A Practical Approach Using CVIPTools.

[25] G. Farnebäck, ''Two-frame motion estimation based on polynomial expansion,'' in Scandinavian conference on Image analysis, vol. 2749. Berlin, Germany: Springer, Jun. 2003, pp. 363–370

[26] Poudel, R. P., Liwicki, S., & Cipolla, R. (2019). Fast-scnn: Fast semantic segmentation network. In: Proc. British Machine Vision Conference.

[27] Q. Liu, Z. He, X. Li, and Y. Zheng, ''PTB-TIR: A thermal infrared pedestrian tracking benchmark,'' IEEE Trans. Multimedia, vol. 22, no. 3, pp. 666–675, Mar. 2020.

[28] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 4293-4302, doi: 10.1109/CVPR.2016.465.

[29] F. Li, C. Tian, W. Zuo, L. Zhang and M. -H. Yang, "Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4904-4913, doi: 10.1109/CVPR.2018.00515.

[30] Y. Song *et al*., "VITAL: VIsual Tracking via Adversarial Learning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8990-8999, doi: 10.1109/CVPR.2018.00937.

[31] X. Li, C. Ma, B. Wu, Z. He and M. -H. Yang, "Target-Aware Deep Tracking," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 1369-1378, doi: 10.1109/CVPR.2019.00146

[32] H. A. P. Bloom, "An efficient filter for abruptly changing systems", in Proceedings of the 23rd IEEE Conference on Decision and Control Las Vegas, NV, Dec. 1984, 656-658