sensors

MDPI

Article

Spatial–Semantic and Temporal Attention Mechanism-Based Online Multi-Object Tracking

Fanjie Meng [1], Xinqing Wang [1,*], Dong Wang [1], Faming Shao [1] and Lei Fu [2]

[1] Department of Mechanical Engineering, College of Field Engineering, Army Engineering University of PLA, Nanjing 210007, China; beilimeng1992@163.com (F.M.); dyhkxywangdong@163.com (D.W.); shaofaming@163.com (F.S.)

[2] Department of Armament Science and Technology, College of Field Engineering, Army Engineering University of PLA, Nanjing 210007, China; fulei10@mails.jlu.edu.cn

* Correspondence: wangxqprof@163.com; Tel.: +86-187-6168-3665

Received: 16 December 2019; Accepted: 10 March 2020; Published: 16 March 2020

Abstract: Multi-object tracking (MOT) plays a crucial role in various platforms. Occlusion and insertion among targets, complex backgrounds and higher real-time requirements increase the difficulty of MOT problems. Most state-of-the-art MOT approaches adopt the tracking-by-detection strategy, which relies on compute-intensive sliding windows or anchoring schemes to detect matching targets or candidates in each frame. In this work, we introduce a more efficient and effective spatial–temporal attention scheme to track multiple objects in various scenarios. Using a semantic-feature-based spatial attention mechanism and a novel Motion Model, we address the insertion and location of candidates. Some online-learned target-specific convolutional neural networks (CNNs) were used to estimate target occlusion and classify by adapting the appearance model. A temporal attention mechanism was adopted to update the online module by balancing current and history frames. Extensive experiments were performed on Karlsruhe Institute of Technologyand Toyota Technological Institute (KITTI) benchmarks and an Armored Target Tracking Dataset (ATTD) built for ground-armored targets. Experimental results show that the proposed method achieved outstanding tracking performance and met the actual application requirements.

Keywords: deep learning; video processing; spatial-temporal attention; multi-object tracking; autonomous vehicle

1. Introduction

Multi-object tracking (MOT) is one of the most fundamental capabilities of unmanned aerial vehicles (UAV), armored scout cars (ARSVs) and other platforms [1,2]. Among them, MOT based on digital image sensing has become a hotspot for research, as it would allow dynamic environments to be captured through an accurate tracking the movement of multiple targets object. Most existing multi-object tracking approaches adopt a two-step procedure. In the first step, the potential candidates are located using a detection algorithm. In the next step, the potential candidates are estimated and linked across different frames. The challenges of multi-object tracking can be summarized as following:

- A tracking system is required to deal with occlusion and insertion. The digital image sensor has a limited receptive field, which means that occlusion and insertion are common. In a receptive field, targets entering and leaving result in boundary insertions, and the close spatial positions of targets in the field result in the occlusion of targets.
- The ability to track small targets is highly important. Small targets are very common in real-life situations, and the ability to recognize small targets gives tracking system a longer response time. This is quite a significant challenge for conventional tracking-by-detection strategies.

- The tracking system is required to be robust. Some scenes, including jungle, desert, and grassland, are more complicated than general scenarios. Dust caused by movement adds complexity to a background.
- There is no ready-made MOT dataset available for armored targets. Compared with general multi-object tracking, the tracking of multi-armored targets is more challenging after adopting camouflage or smoke shielding to avoid exposure.

Figure 1 presents typical frames including vehicles in KITTI tracking benchmarks and armored targets from our Armored Target Tracking Dataset (ATTD).

Figure 1. Typical frames including vehicles in KITTI tracking benchmarks and armored targets from our Armored Target Tracking Dataset (ATTD).

In consideration of the challenges mentioned above, we proposed an online multi-object tracking method based on a spatial–temporal attention mechanism (STAM) [3]. In order to reduce computation, we proposed an Offline Candidates Recommendation Module that was based on a novel spatial-attention map, leveraging semantic features to determine suspect areas as opposed to the sliding windows and dense anchoring scheme in STAM. This strategy can filter out 80% of the invalid areas while maintaining the same recall rate. Considering the irregular movement of armor targets, a novel Motion Model was proposed to analyze the motion trajectories of history frames and predict the precise current position of target. Online-trained target-specific convolutional neural networks (CNNs) were used to estimate the classification and occlusion for each candidate in the same manner as STAM. In order to balance the effects of current and history frames during online training, a temporal attention mechanism was introduced to update the parameters of the target-specific CNNs. Finally, aiming to establish a ready-made MOT dataset for armored targets, we built an Armored Target Tracking Dataset (ATTD) via actual data collection and network downloading. Several experiments were conducted to verify the proposed method on the vehicle-target dataset KITTI and the armored-target dataset ATTD.

Our main contributions are summarized as follows. Firstly, an Offline Candidates Recommendation Module based on a spatial attention mechanism was proposed that could produce fewer false negatives and greatly reduce the computation. Secondly, a novel Motion Model was proposed to locate which candidates provide a full consideration to the possible motion of the target and fit more complex movements. Thirdly, an Armored Target Tracking Dataset (ATTD) was built to address the lack of a ready-made MOT dataset for armored targets.

The rest of our paper is organized as follows: In Section 2, we introduce the related work. In Section 3, we provide an overview of the method then present the details of our multi-object tracking method. The experimental evaluation is provided in Section 4. Finally, we present some conclusions and suggest future work in Section 5.

2. Related Work

2.1. Single-Object Tracking

Tracking is a fundamental task in any video processing that requires some degree of reasoning about objects of interest [4–6]. The methods of object tracking can be divided into two categories: the method suitable for single-object tracking [7–9], and the method suitable for multi-object tracking [10–12]. Until very recently, the most popular single-object tracking method trained a discriminative classifier online (then updated online) using ground-truth information from the first frame to achieve target tracking. The appearance of a target is often the only link to a video frame. These discriminative classifiers usually have a filter- or deep-neural-network-based structure. A few years ago, Bolme et al. [13] proposed Correlation Filtering, a simple algorithm that permits discrimination between the template of an arbitrary target and its 2D translations, to quickly distinguish a single object from the background. Correlation Filtering and its improved tracking method [14–17] are widely used in various tracking applications. However, the Correlation Filtering method has poor performance in tracking targets with obvious deformations. Recently, with the great success of deep convolutional neural networks (CNNs) [18–22], a discriminative offline classifier represented by the Siamese [9] model has been widely applied in cases of single-object tracking. During testing, the Siamese model formulates tracking as a convolutional feature cross-correlating between a target template and a search region. Wang et al. [8] improved the offline training procedure of the popular fully convolutional Siamese approach for single-object tracking by augmenting their loss with a binary segmentation task. Li et al. [7] used comprehensive theoretical analysis and experimental validations to break the Siamese tracker's restriction against deep networks and take advantage of features. They integrated deep networks into the Siamese network to make the network more robust. In conclusion, single-object tracking focuses on tracking the contours of a target to determining its center position.

2.2. Multi-Object Tracking

As opposed to single-object tracking, the core topics being researched for multi-object tracking are the occlusion of multiple targets, the insertion of new targets around boundaries and the disappearances of targets from a scene [23–25]. The appearance cues of obscured targets used for training are polluted when the spatial positions of targets are too close together in a scene. In these cases, a single-object tracker will update the appearance model with the corrupted samples and gradually drift to the occluder [3]. Furthermore, single-object trackers cannot deal with a new target being inserted in the receptive field, and a new ground-truth needs to be added, which is difficult to achieve. At present, most state-of-the-art MOT methods adopt the strategy of tracking-by-detection [11,26–28], which is a two-step procedure composed of a detection module and a tracking module. In the detection module, candidates are recommended in each frame. Then, the candidates are estimated in tracking module. In Son et al. [29], a quadruplet convolutional neural network is proposed for multi-object tracking that can learn to associate object detections across frames using quadruplet losses. Dawei et al. [11] proposed a multi-scale object detector to augment the Single-Shot multi-box Detector (SSD) with temporal regions of interests (ROIs). However, the spatial–temporal relationship of targets is not involved in their method. Chu Q et al. [3] used a spatial–temporal attention mechanism to track multiple objects. They built a Motion Model based on the correlation between current and history frames to recommended candidates. The spatial attention mechanism is used to estimate the occlusion and the temporal attention mechanism is used to realize the online update of the tracking module.

The insertion of new targets around boundaries is disregarded in this work, and the advantages of deep convolutional neural networks are not applied. In addition, a linear Motion Model cannot be applied to a complicated motion.

3. Proposed Method

3.1. Overview of Our Method

With a focus on multi-object tracking, we proposed a multi-object tracking method based on the spatial–temporal attention mechanism shown in Figure 2. The MOT method consists of four parts: (a) an Offline Candidates Recommendation Module; (b) an Online Candidates Estimation Module; (c) a Motion Model; and (d) Temporal Attention Model. Firstly, the current frame is sent into the Offline Candidates Recommendation Module, which is trained offline to predict the suspect areas of candidates. Replacing the sliding window or dense anchoring scheme, we use a spatial-attention map to filter out most areas (such as sky and grass) that are irrelevant to the targets of interest in the offline module. Meanwhile, the insertion of new target around the boundary is solved. Next, a novel Motion Model analyzes the motion curve of the target in the history frames and assists the offline module to determine the location and shape of the bounding box. The ROI features of the candidates are determined and sent to the Online Candidates Estimation Module. In the online module, estimation is operated for each ROI feature, including classification and occlusion. The Temporal Attention Model updates the Online Candidates Estimation Module by balancing the positive and negative samples of the history and current frames. Finally, the multi-object position and classification results of the current frame are evaluated and the model is updated.

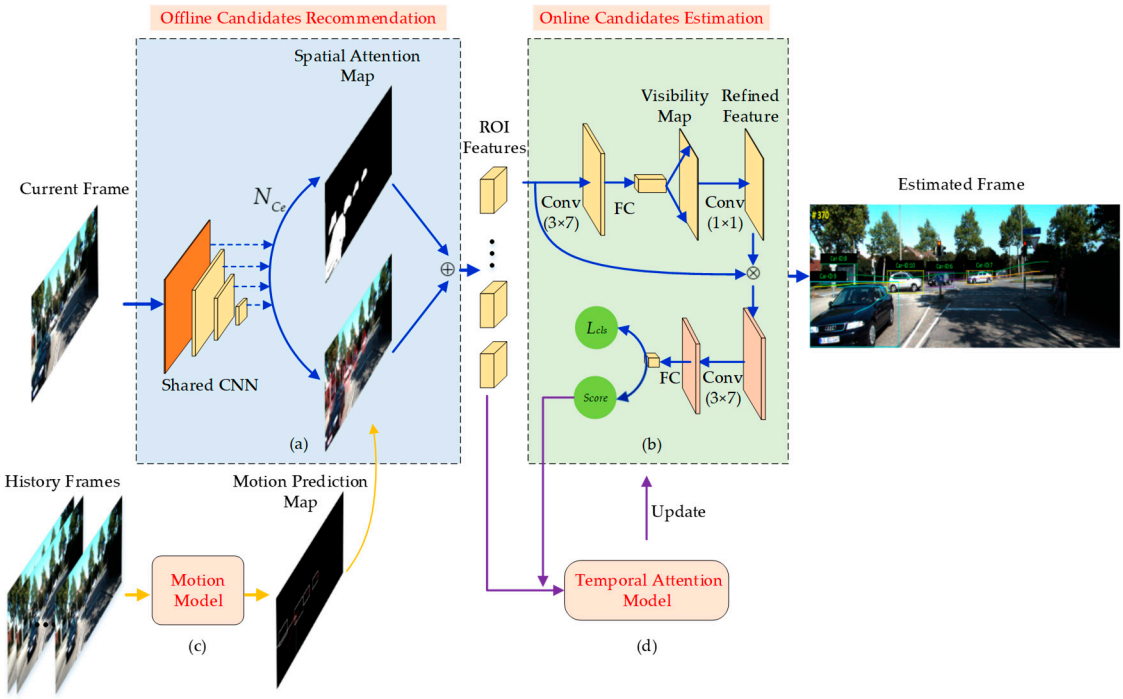

Figure 2. Overview of our method. (**a**) Offline Candidates Recommendation Module; (**b**) Online Candidates Estimated Module; (**c**) Motion Model; (**d**) Temporal Attention Model. $\bigoplus$ demotes combination.

3.2. Offline Candidates Recommendation Module

Detection is the cornerstone of multi-object tracking methods based on tracking-by-detection. With the great success of detectors based on deep convolutional neural networks, offline modules have been widely applied in detecting the stages of multi-object tracking. Among them, region proposal networks (RPNs) [30] are considered to be the most successful ROI proposal method, and are widely used in many detection applications [31–35]. In an RPN, anchors are defined as a set of sliding windows with fixed scales and aspect ratios [30]. In order to ensure a sufficiently high recall for proposals, a large number of anchors are used in such methods. Obviously, if this exhaustive strategy is adopted

in two-step multi-object tracking, the process of estimating large numbers of candidates is extremely computation-expensive. The main reason for this is that most of the bounding box (or anchors) are placed in areas that do not contain targets. Inspired by Wang et al. [36], we adopt a spatial-attention map to recommend the candidates at the detection stage. As shown in Figure 3, our Offline Candidates Recommendation Module includes a shared features extraction CNN, a spatial-attention branch N_s and a Motion Model and bounding box prediction branch N_m.

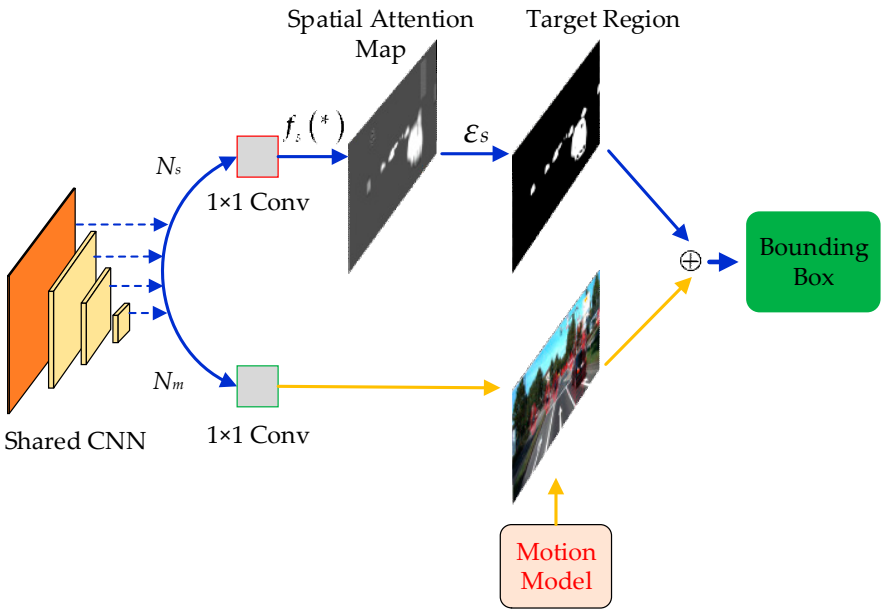

Figure 3. Offline Candidates Recommendation Module.

A smaller number of candidates containing all targets means a decrease in calculations of regression and classification. We present an effective and efficient scheme that leverages semantic features to guide the bounding box. As shown Figure 3, we use a spatial-attention branch N_s and each feature map $\mathrm{F_I}$ to generate a spatial-attention map $\mathrm{M_s}$ of the target, which can be formulated as

$$\mathrm{M_s} = \mathrm{M}(\cdot|\mathrm{F_I}) = f_s(\mathrm{F_I}; \omega_s), \mathrm{M}(\cdot|\mathrm{F_I}) \in \mathbb{R}^{w \times h}, \mathrm{F_I} \in \mathbb{R}^{w \times h}, \tag{1}$$

where, ω_s is the set of parameters in the spatial-attention branch N_s, and $f_s(*)$ is modeled as a 1×1 convolution with an element-wise sigmoid function. Each $\mathrm{M}(i, j|\mathrm{F_I})$ corresponds to the location with coordinate $\left(\left(i + \frac{1}{2}\right)s, \left(j + \frac{1}{2}\right)s\right)$, where s is the stride of the feature map. For each location aim-listed spatial-attention values, we adopt a global threshold ε_s to determine whether the location belongs to a target, which can be formulated as

$$\begin{cases} \mathrm{M}(i, j|\mathrm{F_I}) \geq \varepsilon_s, & \textit{Target}; \\ \mathrm{M}(i, j|\mathrm{F_I}) < \varepsilon_s, & \textit{Background}. \end{cases} \tag{2}$$

According to the spatial-attention values of each position and the global threshold value ε_s, we determine the active regions where targets may possibly exist. This process can filter out about 80% of the regions while still maintaining the same recall. The determination of the bounding box shape and center location are introduced in the next chapter. Figure 4 shows an example of a spatial-attention map generated by the branch N_s and 3D probability features of targets. In the spatial-attention map, the 3D probability features of targets are more prominent than background.

(a) (b) (c)

Figure 4. Example of the spatial-attention map and 3D spatial-attention value features of targets. (**a**) Input frames; (**b**) spatial-attention map; (**c**) 3D spatial-attention value features of targets.

3.3. Motion Model

A motion model analyzes the motion curve of the target in the history frames and predicts the position of the target in the current frame [37,38]. Most single-object trackers do not use a motion model [6–9]. However, the motion model has proven helpful in multi-object tracking, which can help locate targets and realize the correspondence of multi-target labels in different frames. In most MOT applications [29,39–41], a simple linear motion model is used to estimate the target state. Such motion models may cause a loss of tracking when the target turns quickly, suddenly stop or drives in reverse. In this work, we give full consideration to the possible motions of a target and propose a novel Motion Model to locate the candidates. In the spatial-attention branch N_s, we determine the possible areas of bounding boxes. The final position of candidates is determined with branch N_s and the Motion Model. The estimated state of *k_th* candidate C^k at t frame can be formulated as

$$X_t^k = \left[x_t^k, y_t^k, w_t^k, h_t^k\right], \tag{3}$$

where x_t^k and y_t^k represent the center location of the candidate, and w_t^k and h_t^k denote the width and height of candidate, respectively. In our Motion Model, the predicted state set Q_{t+1}^k of C^k at $t+1$ frame can be expressed as

$$\begin{aligned} Q_{t+1}^k &= \left\{\widetilde{X_{t+1,n}^k}\right\}_{n=1}^{8} \\ &= \left\{X_t^k + v_t^k([\pm 1,0,0,0]^{\mathrm{T}}, [0,\pm 1,0,0]^{\mathrm{T}}, \tfrac{1}{2}\left[\pm\sqrt{2}, \pm\sqrt{2}, 0, 0\right]^{\mathrm{T}}, \tfrac{1}{2}\left[\pm\sqrt{2}, \mp\sqrt{2}, 0, 0\right]^{\mathrm{T}})\right\} \end{aligned} \tag{4}$$

where $\widetilde{X_{t+1,n}^k}$ is the *n_th* predicted state of candidate C^k at frame $t+1$, and v_t^k represents the velocity of *k_th* candidate C^k at frame t. Figure 5a shows the spatial positions of the relative predicted candidates at frame $t+1$. In order to cover the possible motion of the target, eight predicted states are used to formulated the candidates, which divide the direction of the space equally. Figure 4b shows an example of the response of the same target's candidate to the motion model at different frames. In the figure, the green arrow represents the speed of the target, the blue dotted box represents the target bounding box at frame t, and the red box represents the target bounding box at frame $t+1$. The response of target C^1 is $X_{t+1,6}^1$ with a velocity v_t^1. In the same way, the response of target C^2 is $X_{t+1,1}^2$ with a velocity v_t^2, and the response of target C^3, C^5 and C^6 is $X_{t+1,5}^k$ (k = 3, 5, 6) with a velocity v_t^3, v_t^5 and v_t^6, respectively. For target C^4 with sudden turning, the orange dotted box represents the predicted position in the linear motion model and the red box represents the predicted position in our motion model. Obviously, our motion model had the better prediction ability, but a linear motion model cannot be applied to all situations.

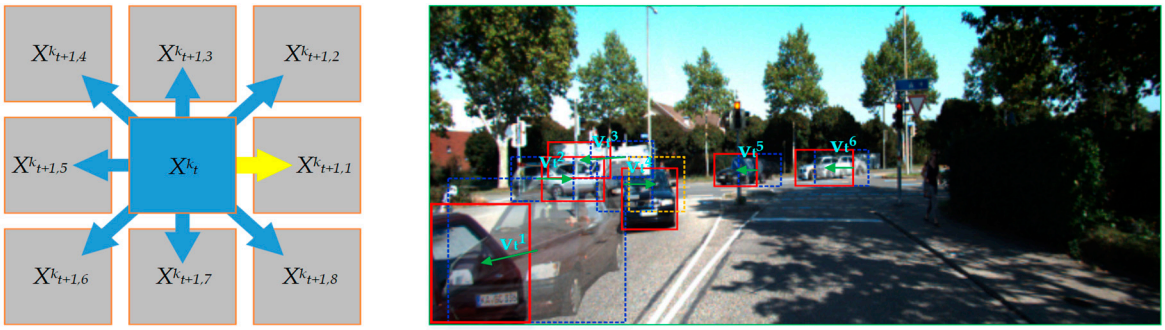

Figure 5. (**a**) Spatial positions of the relative predicted candidates at frame $t+1$; (**b**) example of the response of the same target's candidate to the motion model at different frames.

In the spatial-attention branch N_s, a spatial-attention map $\mathrm{M_s}$ is used to predict active regions where targets may possibly exist. We count the number of points that are greater than the threshold ε_s in each predicted position and select the position with the most points that satisfies above condition as the response. The response process can be expressed as

$$X^k_{t+1} = arg \underbrace{\max}_{X^k_{t+1,n} \in M^k_{t+1}} \sum_{\mathrm{M}(i,j|X^k_{t+1,n}) \geq \varepsilon_s} \mathrm{M}\left(i, j \middle| X^k_{t+1,n}\right),\ n = 1, 2, \ldots, 8. \tag{5}$$

Considering the occlusion, we take the direction of v^k_t as the main direction. In this case, X^k_{t+1} is formulated as

$$X^k_{t+1} = X^k_t + \left[v^k_t, 0,\ 0\right], \tag{6}$$

where $v^k_t = \frac{1}{T_{t+1}-T_t}([x^k_{t+1},\ y^k_{t+1}]^T - [x^k_t,\ y^k_t]^T)$ and $[x^k_t,\ y^k_t]$ represent the center of candidate.

3.4. Online Candidates Estimation

Different from single-object tracking, the core research of multi-object tracking includes the occlusion of multiple targets, the insertion of new targets around boundaries and the disappearances of targets in the scene area. Occlusion is an important cue that needs to be considered during the online updating process. The appearance features of targets are polluted and cannot be used as online update samples when they are occluded by another target, building, fire or smoke. However, in the Motion Model and offline trained classifier, the covered position still scores highly. In this case, the orresponding tracker updates the appearance model with the corrupted samples and gradually drifts to the occluder or background.

The deep features are extracted from shared CNNs using ROI pooling, which ignores the occlusion. In order to address the occlusion, we use target-specific CNNs to estimate the candidates and classify the targets and background. The ROI-pooled feature representation of the *k_th* candidate C^k is denoted as $\Phi_{roi}\left(X^k_t\right) \in \mathbb{R}^{w \times h \times c}$. As in [3], a visibility map X^k_t is output to encode the spatial visibility of the input samples, which can be expressed as

$$\mathrm{V}_{vis}\left(X^k_t\right) = f_{vis}\left(\Phi_{roi}\left(X^k_t\right); \omega^k_{vis}\right),\ \mathrm{V}\left(X^k_t\right) \in \mathbb{R}^{w \times h}, \tag{7}$$

where ω^k_{vis} is the set of visibility parameters of the *k_th* target-specific CNN, and $f_{vis}(*)$ is modeled as both a convolutional layer ($kernel\ size = 3 \times 7 \times 32$) and a fully connected layer ($output\ size = w \times h$). We estimate the *k_th* candidate with an occlusion score p^k_t:

$$p^k_t = f_{cls}\left(\Psi_{ref}\left(X^k_t\right);\ \omega^k_{cls}\right),\ p^k_t \in [0, 1], \tag{8}$$

where ω_{cls}^{k} is the set of classification parameters of the *k_th* target-specific CNN, and $f_{cls}(*)$ is modeled as both a convolutional layer (*kernel size* $= 3 \times 7 \times 32$) and a fully connected layer (*output size* $= 1$). $\Psi\left(X_t^k\right) \in \mathbb{R}^{w \times h \times c}$ denotes the refined features of the *k_th* candidate C^k, which is expressed as

$$\Psi_{ref}\left(X_t^k\right) = \Phi_{roi}\left(X_t^k\right) \circ f_{con}\left(V_{vis}\left(X_t^k\right); \omega_{con}^k\right) \Psi_{ref}\left(X_t^k\right) \in \mathbb{R}^{w \times h \times c} \quad (9)$$

where $\circ$ represents the channel-wise Hadamard product operation, $f_{con}(*)$ denotes a local connected layer with a spatial SoftMax layer, and ω_{con}^k is the set of connected parameters of the *k_th* target-specific CNN.

Figure 6 shows examples of occlusion and generated visibility maps. The last column shows that the classification score is lower when the target is occluded by the background. However, when the target is occluded by a same-class target, the classification score is able to classify neither the tracked target nor the others. In the generated visibility maps, the degree of target occlusion is well evaluated, even if it is occluded by the same-class target. In this work, we use a threshold p_0 to estimate the degree of target occlusion. *k_th* candidates are taken as the tracking target without occlusion when $p_t^k \geq p_0$. On the contrary, *k_th* candidates are taken as an occluded target when $p_t^k < p_0$. p_0 is a classification threshold.

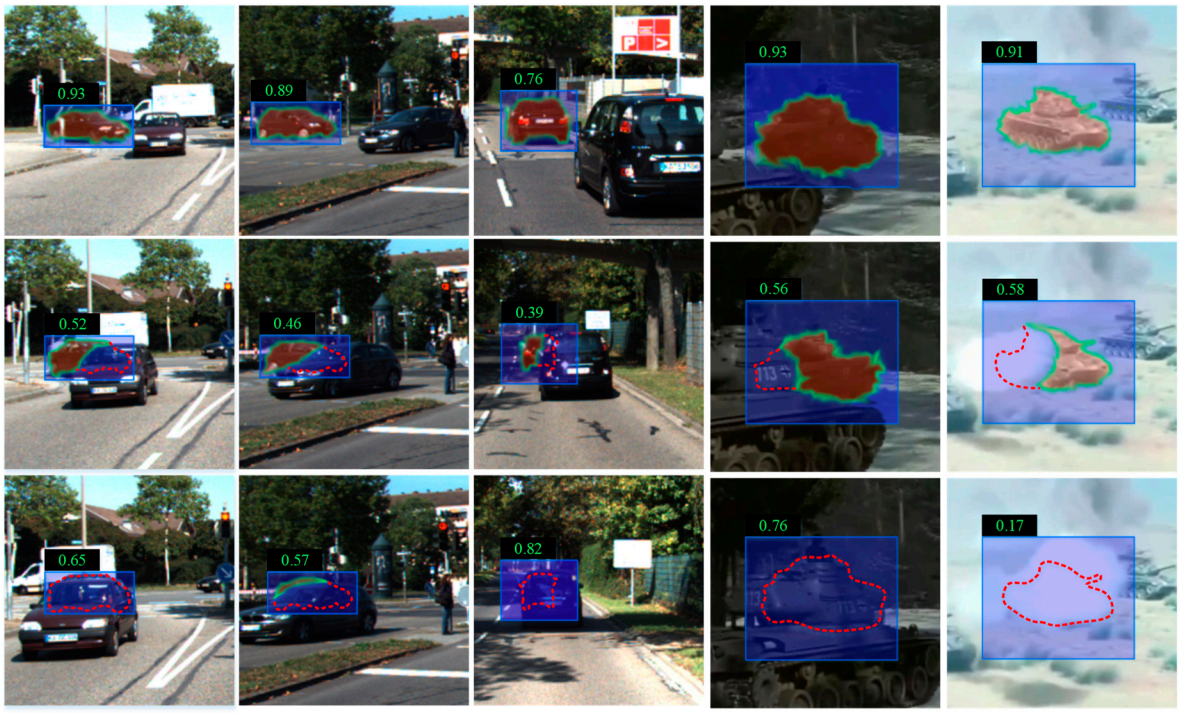

Figure 6. Examples of occlusion and generated visibility maps. The dotted lines surround the occluded targets. The blue bounding box is the predicted state set in the Motion Model. The numerical value is the classification score of candidates in the offline trained classifier.

3.5. Temporal Attention Model

The polluted features of corrupted samples in bounding boxes would reduce the ability of estimation model to classify targets and backgrounds until the candidates cannot be evaluated. To address this conflict, a Temporal Attention Model is introduced in this work to balance the history and current frames in the online training process. As shown in Figure 2, the positive samples in history frames are saved in the Temporal Attention Model according to the scores of candidates in the estimation model. The history positive sample refers to a sample whose classification score is larger than the classification threshold ($p^k \geq p_0$), which reflects the original visual features of the target while the positive sample in current frame reflects the change of the visual features.

In this work, the Temporal Attention Model is used to update the Online Candidates Estimate Module and preserve historical positive samples. When the classification score of a candidate is larger than the classification threshold ($p^k \geq p_0$), the target is successfully tracked, and positive samples are used to update the online module—including the candidate of the same target in the current frame and the historical candidate in the Temporal Attention Model. The negative samples are all selected randomly from the current frame except for the candidate's region. When the classification score of a candidate is lower than the classification threshold ($p^k < p_0$)—that is, the target is untracked—the positive samples used to update the online module all come from the Temporal Attention Model and the negative samples are all selected randomly from the current frame. For candidate C^k, the target-specific loss function in t frame can be expressed as

$$L = L_t^{k-} + \lambda L_t^{k+} + (1-\lambda) L_h^{k+}, \tag{10}$$

where λ is a temporal attention parameter to balance the current and history samples, which can be expressed as

$$\lambda = \begin{cases} 0 & p^k < p_0 \\ 0.9 & p^k \geq p_0 \end{cases}, \tag{11}$$

where L_t^{k-} is the loss of negative samples in the current frame, L_t^{k+} is the loss of positive samples in the current frame, and L_h^{k+} is the loss of positive samples in the history frames. L_t^{k-}, L_t^{k+}, and L_h^{k+} can be expressed respectively as

$$L_t^{k-} = -\frac{1}{N_t^{k-}} \sum_{i=1}^{N_t^{k-}} log\Big[1 - f_{cls}\Big(\Psi_{ref}\Big(X_t^{k-}\Big);\ \omega_{cls}^k\Big)\Big], \tag{12}$$

$$L_t^{k+} = -\frac{1}{N_t^{k+}} \sum_{i=1}^{N_t^{k+}} log f_{cls}\Big(\Psi_{ref}\Big(X_t^{k+}\Big);\ \omega_{cls}^k\Big), \tag{13}$$

$$L_h^{k+} = -\frac{1}{N_h^{k+}} \sum_{i=1}^{N_h^{k+}} log f_{cls}\Big(\Psi_{ref}\Big(X_h^{k+}\Big);\ \omega_{cls}^k\Big), \tag{14}$$

where N_t^{k-}, N_t^{k+}, and N_h^{k+} are the number of negative and positive samples in the current frame and positive samples in history frames. In this work, we used a BP algorithm to update the weight parameters of each layer in the online estimation module.

3.6. Training of Module

3.6.1. Training of the Offline Candidate Recommendation Module

In our Offline Candidate Recommendation Module, spatial-attention branch N_s and a Motion Model are used to generate candidates. The offline module is optimized in an end-to-end fashion using a multi-task loss. In order to train the spatial-attention branch, a target location loss L_{loc} is introduced based on Focal Loss [42]. At the initial stage of tracking, a bounding box prediction branch N_m is used to generate a bounding box and solve the problem of generating the Motion Model. The conventional classification loss L_{cls} and regression loss L_{reg} are adopted to train the branch N_m. The two branches are jointly optimized with the following loss:

$$L = L_{loc} + L_{cls} + L_{reg} \tag{15}$$

In the training of spatial-attention branch N_s, binary labeled maps are used as samples. In the binary labeled maps, 1 represents the valid location of a target center and 0 represents an invalid

location. In this work, ground-truth bounding boxes are used to guide the generation of samples. Let $\left(x_g, y_g, w_g, h_g\right)_n$ represent the mapped ground-truth bounding box in *n_th* feature map. The center region in the binary labeled map can be expressed as

$$R_{center} = \left(x_g, y_g, 0.1w_g, 0.1h_g\right) \tag{16}$$

The invalid region is the feature map excluding the mapped ground-truth bounding box, which can be expressed as

$$R_{invalid} = F_n \setminus \left(x_g, y_g, w_g, h_g\right)_n \tag{17}$$

where F_n represents the *n_th* feature map.

3.6.2. Training of the Online Candidate Estimation Module

In Online Candidate Estimation Module, we use target-specific CNNs to estimate the candidates and classify the target and background. The target-specific CNNs predict occlusion scores to estimate the candidates. At the initial stage of tracking, the parameters of the target-specific CNNs are random, and the networks have no estimation ability. In order to train the online module, we take the detection results of the Offline Candidate Recommendation Module as positive samples. Let $(x_d, y_d, w_d, h_d)_k$ represent the *k_th* detection result and positive sample. Negative samples are constructed by positive samples and position offset. The *k_th* negative sample according to the positive sample can be formulated as

$$Samples_N = \left\{(x_d \pm \sigma_1 w_d, y_d \pm \sigma_2 h_d, w_d, h_d)_k, (x_d \pm \sigma_1 w_d, y_d \mp \sigma_2 h_d, w_d, h_d)_k\right\}, \tag{18}$$

where σ_1 and σ_2 are randomly selected within the interval $[0.7, 0.9]$. In order to achieve a robust performance, the online target-specific CNNs need sufficient samples to be trained. Denote the frame rate of the video as N_v and use $N_{init} = 0.2N_v$ to complete the training of the Online Candidate Estimation Module once the frames are sufficient. In actual training we use the first 20 frames to complete the training of the online module when the number of video frames is less than 100.

4. Experiment

4.1. Dataset and Implementation

In the multi-object tracking task, an initially unknown number of targets must be tracked as bounding boxes in a video. At present, multi-object tracking (MOT) datasets for general targets like pedestrians and vehicles have been published. The MOT Challenge datasets [43,44] show pedestrians from a variety of different viewpoints. The KITTI tracking dataset [45] features video from a vehicle-mounted camera and consists of 21 training sequences and 29 test sequences. However, such datasets do not contain particular armored targets or complex battlefield scenes (Figure 1). In this work, we built a dataset for armored targets, named the Armored Target Tracking Dataset (ATTD). The ATTD contained 80 (50 training, 30 test) video sequences in a complex battlefield scene, including various battlefield terrains (such as jungle, desert, grassland, and city) and complicated factors (such as armored clustering, muzzle fire and smoke, dust, and so on). All videos were captured by actual shooting and downloaded from the internet. All frames in the ATTD were normalized to a size of 1920×770 pixels. Armored target scales in the ATTD had wide range from 10×10 pixels to more than 700×700 pixels, with an emphasis on remote, small-armored targets. In this work, we use the KITTI training set and the ATTD to evaluate our MOT method for vehicles and armored targets.

In this work, pre-trained Resnet50 models [40] were used as the backbone network for the Offline Candidate Recommendation Module. The Offline Candidate Recommendation Module was trained with Adam, with a momentum of 0.9 and a weight decay of 0.0005, using a single NVIDIA GeForce GTX 2080ti GPU with 11 GB of memory. The Online Estimation Module was trained with the BP algorithm.

4.2. Evaluation Metrics

To evaluate the performance of our multi-object tracking method, we adopted the widely used CLEAR MOT metrics [46], including multiple-object tracking precision (MOTP) and multiple-object tracking accuracy (MOTA). MOTP represents the total error in estimated position for matched object-hypothesis pairs over all frames, averaged by the total number of matches made, which can be expressed as

$$\text{MOTP} = \frac{\sum_{k,t} d_t^k}{\sum_t N_t} \tag{19}$$

where d_t^k is the distance between the *k_th* center of the ground-truth bounding box and its corresponding hypothesis in frame t. N_t is the total number of matches made. MOTP reflects the ability of the multi-object tracker to estimate precise object positions, independent of its skill at recognizing object configurations, keeping consistent trajectories. MOTA can be seen as derived from three error ratios

$$\text{MOTA} = 1 - (\text{FP} + \text{FN} + \text{IDS}) = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{20}$$

where m_t, fp_t, and mme_t are the number of false negatives (FN), false positives (FP) and identity switches (IDS), respectively. MOTA reflects all object configuration errors, including false positives, misses, and mismatches made by the multi-object tracker over all frames. Additionally, the percentage of mostly tracked targets (MT) and mostly lost targets (ML) are used as metrics in this work.

4.3. The Setting of Parameters

In our MOT algorithm, the birth/death of the trackers was determined by the global center threshold ε_s and classification threshold p_0. The former determines whether a location belongs to a target and the latter determines the classification of armored targets and backgrounds and estimates the occlusion. In order to select appropriate parameters, we conducted an exhaustive experiment on the small training set, where several performance indicators are used for estimation. First, we randomly selected 1000 frames from the videos of the KITTI dataset. Half of the frames were used as training samples and the other half belonging to the same video were taken as test samples. Then, the prediction accuracy of the bounding box (Box Accuracy) of the Offline Candidates Recommendation Module was used to evaluate the threshold ε_s. The classification accuracy of positive samples (tracked targets) and negative samples (background, occlusion) in the Online Candidates Estimation Module was used to evaluate the threshold p_0. Meanwhile, we used MOTA as the joint performance indicator of ε_s and p_0. The experimental results are shown in Figure 7. Figure 7a shows the variation of prediction accuracy of the bounding box (Box Accuracy) with the global threshold ε_s and the result of classification accuracy of the target determined by the classification threshold p_0. When ε_s lies in 0.65~0.9, the Offline Candidates Recommendation Module has a higher Box Accuracy. When p_0 lies in 0.5~0.7, the classification accuracy of the Online Candidates Estimation Module is higher. Therefore, in the above two intervals, we selected the appropriate threshold ε_s and p_0 through the MOTA of the whole algorithm. Figure 7b shows the MOTA of our algorithm with different global threshold ε_s and classification threshold p_0. The results demonstrate that the algorithm achieved the highest MOTA (83.5%) on the selected samples, when $\varepsilon_s = 0.7$ and $p_0 = 0.65$. Hence, the next experiments were performed under the values listed above.

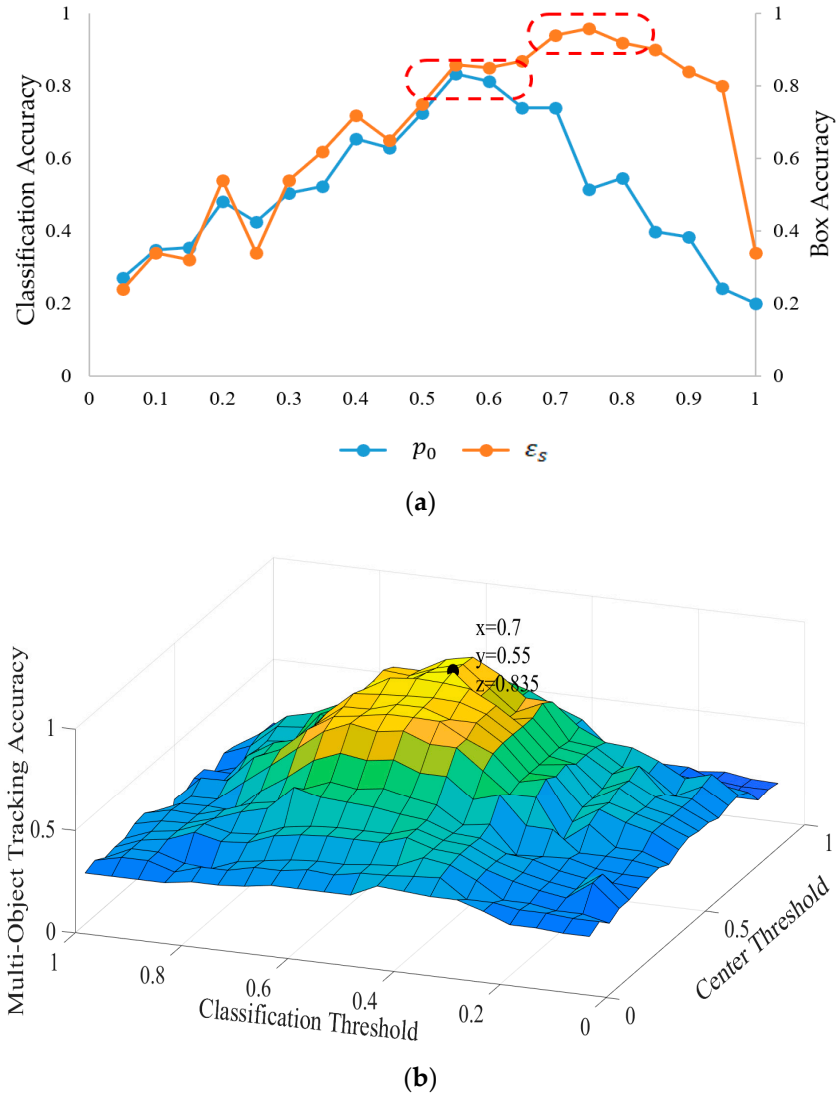

Figure 7. The selection of the global threshold ε_s and classification threshold p_0. (**a**) The relationship between the global center threshold ε_s and the predicted bounding box accuracy. The result of classification accuracy of armored targets determined by the classification threshold p_0. (**b**) The result of MOTA at different values of the global center threshold ε_s and classification threshold p_0.

4.4. Analysis of Candidates Recommendation

In most MOT methods, an anchor-based RPN is the cornerstone of the object detection step. In order to ensure a sufficiently high recall for proposals, a large number of anchors are used in traditional detectors. Obviously, this scheme is extremely waste-computed. In this work, we adopted an alternative strategy to filter out most areas that were irrelevant to the objects of interest such as sky, grassland, desert and so on. In our MOT method, the ROI features of candidates were recommended by the Offline Candidates Recommendation Module and Motion Model. We used a spatial-attention branch N_s and each feature map F_I to generate a spatial-attention map M_s of the target. A global threshold ε_s was used to determine whether a location belonged to a target. The accurate positions of bounding boxes were further determined by assistance of the Motion Model. In order to demonstrate the ability of our Offline Candidates Recommendation Module and Motion Model, we studied the IoU distribution of proposals generated by three algorithms with different components. The details of each algorithm are described as follows:

M1: The shared features extraction CNN + "RPN + 9 anchors";

M2: The shared features extraction CNN + N_s + ε_s + "RPN + 9 anchors";
M3: The shared features extraction CNN + N_s + ε_s + Motion Model.

"RPN + 9 anchors" entails using three scales and three aspect ratios at each feature level. Figure 8 shows the IoU distribution of the three algorithms. The recommendation ability of the probability map is better than that of the RPN (M3 ≈ M2 > M1) when the IoU is set in a higher range (> 0.8). Meanwhile, the number of proposals in M3 are significantly lower than the other two methods when the IoU is set in 0.5 ~ 0.75. The reason for this is that using the spatial-attention map M_s and global threshold ε_s can result in filtering out most areas that are irrelevant to the object. In both datasets, the number of targets contained in each frame is generally 3 ~ 5. Obviously, the excessive proposals of the RPN are redundant.

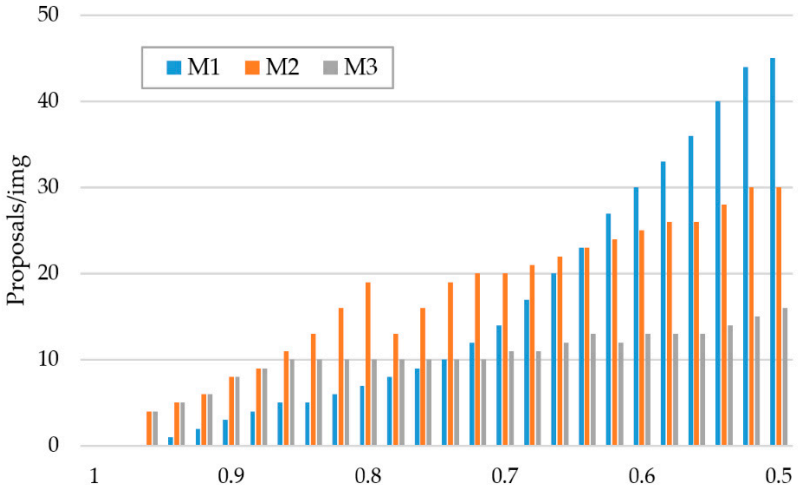

Figure 8. The IoU distribution of the M1, M2 and M3 proposals.

In order to further verify our candidate recommendation method, we compared MOTA, MOTP, MT, and ML for M1, M2 and M3 using the KITTI dataset, which is shown in Table 1. The comparison results of M1 and M2 show that, compared with a traditional RPN, our bounding box center prediction branch N_s improved MOTA, MOTP, and MT by 10.85%, 1.13%, and 7.92%, respectively. The reason for this is that the branch N_s filters out most areas that do not contain the target and reduces background interference. The comparison results of M3 and M2 show that our candidates recommendation method improved MOTA, MOTP, and MT by 13.13%, 14.13%, and 11.37%, respectively. This significant improvement clearly suggests that our Offline Candidates Recommendation Module and Motion Model are crucial for the detection step in multi-armored target tracking.

Table 1. Analysis of the Offline Candidates Recommendation and Motion Model using the KITTI dataset.

Method	MOTA	MOTP	MT	ML
M1(Shared CNN + RPN + 9 anchors)	45.47%	65.30%	27.54%	19.35%
M2(M1 + N_s + ε_s)	56.32%	66.43%	35.46%	18.36%
M3(Shared CNN + N_s + ε_s + Motion Model)	69.45%	80.56%	46.83%	17.25%

4.5. Analysis of Candidates Estimation

In our MOT method, the Online Candidates Estimation Module classifier recommended ROI features into targets and backgrounds and estimated the occlusion. The Temporal Attention Model saves history positive samples and updates the online module. The polluted features of corrupted samples in bounding boxes can reduce the ability of model estimates to classify targets and backgrounds until the candidates cannot be evaluated. In order to prevent the degradation of target-specific CNNs

in the Online Candidates Estimation Module, we used the Temporal Attention Model to balance the current and history frames. In order to verify the validity of our Online Candidates Estimation Module and Temporal Attention Model, a comparison experiment involving three algorithms was conducted. The details of each algorithm are described as follows:

M3: The shared features extraction CNN + N_s + ε_s + Motion Model;
M4: M3 + Online Candidates Estimation Module;
M5: M3 + Online Candidates Estimation Module + Temporal Attention Model.

The comparison results are shown in Table 2. Compared with use of offline module only (M3), M4 improved MOTA by 15.87%, which demonstrates that our Online Candidates Estimation Module can effectively distinguish tracking targets from backgrounds. Compared with M4, M5 improved MOTA by 4.33%. The reason for this is that our Temporal Attention Model prevents the degradation of target-specific CNNs by balancing the history and current frames, which is illustrated in the significant reduction in ML by 13.11%. Meanwhile, MOTP and MT were also improved by 4.03% and 5.19% respectively.

Table 2. Analysis of the Offline Candidates Recommendation and Motion Model using the KITTI dataset.

Method	MOTA	MOTP	MT	ML
M3(Shared CNN + N_s + ε_s + Motion Model)	69.45%	80.56%	46.83%	17.25%
M4(M3 + Online Candidates Estimation)	85.32%	81.52%	75.46%	15.36%
M5(M4 + Temporal Attention Model)	89.65%	85.55%	80.65%	2.25%

Figure 9 shows an example of one target occluded by another when they are close to each other. The target-specific tracker gradually drifts to the occluder without the Temporal Attention Model. The classification score of target 33 decreases gradually, until it is occluded. The polluted features of target 33 gradually drift its bounding box to target 31.

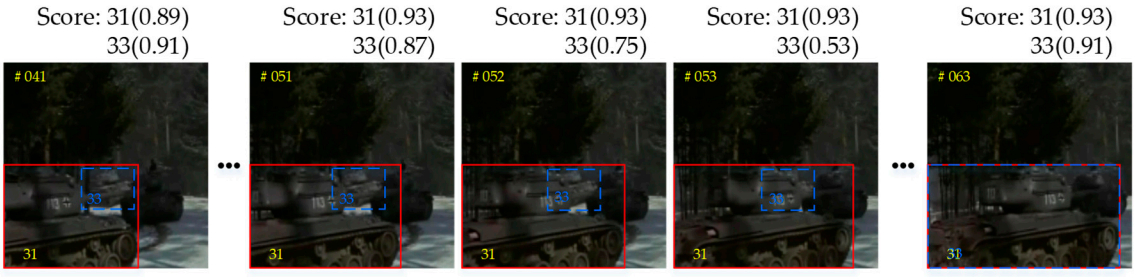

Figure 9. An example of drift caused by occlusion without the Temporal Attention Model.

4.6. Benchmark Evaluation Results

To demonstrate the effectiveness of our online multi-object tracking method, we compared our algorithm to several state-of-the-art approaches using both the KITTI and ATTD, including offline tracking methods like Siamese CNN [47], Convolutional Neural Networks and Temporally Constrained Metrics (CNNTCM) [48], Discrete-Continuous Energy Minimization (DCO-X) [49], and Learning Optimal Structured Support Vector Machine LP-SSVM [50], Online tracking methods like Near-Online Multi-Target Tracking (NOMT-HM) [51], Structural Constraint Event Aggregation (SCEA) [52], Spatial-Temporal Attention Mechanism (STAM) [3], Successive Shortest Path (SSP) [53], multi-modality Multi-Object Tracking (mmMOT) [54], and Multi-Object Tracking Beyond Pixels (MOTBeyondPixels) [55]. Some of these approaches could only be performed in the offline setting. We reimplemented these methods into our platform and used the average tracking time per frame as the tracking time.

Results using the KITTI dataset: The comparison results with the ATTD tracking testing set are summarized in Table 3. Our approach achieved the best MOTP, MT, and ML. Compared with the second best, we obtained 4.88% and 7.42% increases in MOTA and MT, respectively, and a 0.02% decrease in ML. Our method achieved the second best MOTP by 85.55%. Compared with online methods, the offline methods generally had a longer tracking time. Further, in most cases, there were not enough samples to train the offline model in MOT tracking. The higher MOTA and MT indicate that our Offline Candidates Recommendation Module filtered out most areas that did not contain the target, reduced background interference, and was crucial for the detection step. The best MOTP indicates that the Motion Model could accurately predict the trajectory of motion. The decrease in ML indicates that our approach had fewer false negatives, which should be largely attributed to our Temporal Attention Model preventing the degradation of target-specific CNNs by balancing the history and current frames. The overall running time indicates that our approach met the actual requirements.

Table 3. Comparison with state-of-the-art methods using the KITTI tracking testing set.

Method	Mode	MOTA	MOTP	MT	ML	Tracking Time (s)
Siamese CNN	offline	46.31%	71.20%	15.52%	27.30%	0.81
CNNTCM	offline	49.50%	71.80%	19.75%	22.64%	0.73
LP-SSVM	offline	75.65%	77.80%	42.54%	10.25%	0.95
mmMOT	offline	84.77%	85.21%	73.23%	2.77%	0.16
NOMT-HM	online	69.12%	78.52%	38.51%	15.28%	0.09
STAM	online	77.20%	74.90%	29.65%	18.57%	0.25
SSP	online	68.00%	79.52%	42.05%	10.64%	0.61
MOTBeyondPixels	online	84.24%	85.73%	73.23%	2.77%	0.30
Ours	online	89.65%	85.55%	80.65%	2.25%	0.16

Results using the ATTD: The comparison results using the ATTD tracking testing set are summarized in Table 4. Our approach achieved the best MOTP, MT and ML. Compared with the second best, we obtained 3.45%, 4.04%, and 2.71% increases in MOTA, MOTA, and MOTP, respectively, and a 3.98% decrease in ML. Figure 10 shows some qualitative results of our MOT method using both KITTI and ATTD, where we merged two sequential frames to make their difference apparent. The object trajectories are also shown in the figure.

Table 4. Comparison with state-of-the-art methods on the ATTD tracking testing set.

Method	Mode	MOTA	MOTP	MT	ML	Tracking Time (s)
SiameseCNN	offline	46.31%	71.20%	15.52%	27.30%	0.81
CNNTCM	offline	49.50%	71.80%	19.75%	22.64%	0.73
DCO-X	offline	65.12%	73.85%	31.52%	14.25%	0.95
LP-SSVM	offline	75.65%	77.80%	42.54%	10.25%	0.16
NOMT-HM	online	69.12%	78.52%	38.51%	15.28%	0.09
SCEA	online	34.35%	71.10%	47.35%	37.50%	0.18
STAM	online	77.20%	74.90%	29.65%	18.57%	0.25
SSP	online	68.00%	79.52%	42.05%	10.64%	0.61
Ours	online	80.65%	83.55%	49.52%	6.27%	0.26

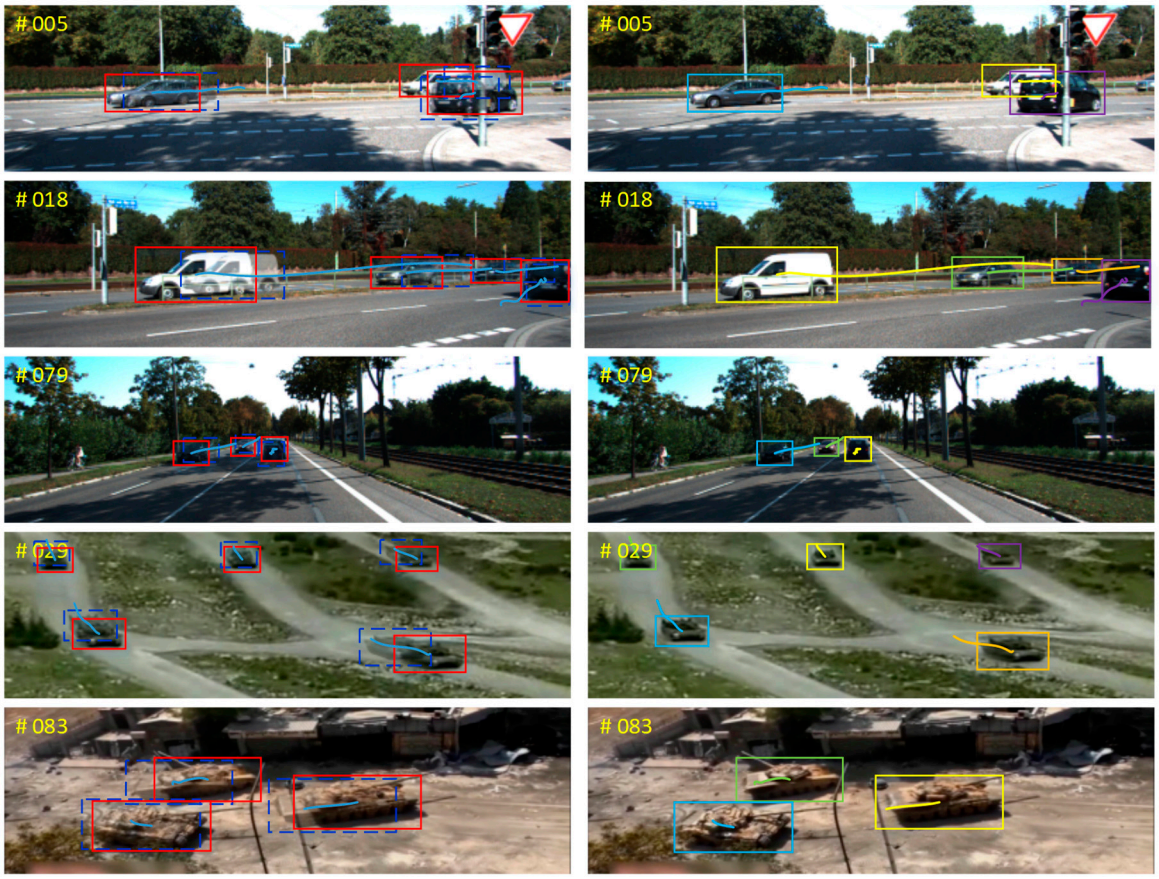

Figure 10. Experiment results using both the KITTI dataset and the ATTD. (**a**) Detection results with our Offline Candidates Recommendation Module. The previous position is denoted with a blue rectangle and the current with a red. (**b**) Tracking results using our MOT method and target trajectories.

5. Conclusions

The tracking of multiple objects can be complicated by occlusion, insertion among targets, complex backgrounds, and real-time requirements. Moreover, there is no ready-made multi-object tracking (MOT) dataset for armored targets. In this work, we proposed an online multi-object tracking method and a special MOT dataset for armored targets, named the Armored Target Tracking Dataset (ATTD). Combining the exhaustive strategy of traditional RPN with new target insertion, we used an Offline Candidates Recommendation Module to recommend candidates in the detection stage. The offline module adopted a spatial-attention branch N_s to filter out most areas that were irrelevant to the objects of interest. A novel Motion Model was proposed to assist in locating the candidates and provide full consideration to the possible motion of the target. Significant improvements in several comparison experiments clearly suggests that our Offline Candidates Recommendation Module and Motion Model were crucial to the detection step in multi-object tracking. In order to address the occlusion among targets, we used target-specific CNNs to estimate the candidates in the Online Candidates Recommendation Module, which estimates target occlusion and classification. In order to prevent the polluted features of corrupted samples from reducing the ability of the estimation model, a Temporal Attention Model was introduced to balance the history and current frames in the online training process. Our Online Candidates Estimation Module could effectively distinguish tracking targets from the background. The Temporal Attention Model prevented the degradation of target-specific CNNs by balancing the history and current frames. Experimental results show that our method achieved outstanding increases in MOTA, MOTP, and MT, and decreased ML. The overall running time indicates that our approach was able to meet the requirements of the experiment. In the future, we will test this

method with other MOT datasets and consider strengthening the detection by merging visible and infrared images.

Author Contributions: Conceptualization, F.M. and X.W.; Methodology, F.M. and X.W.; Software, F.M.; Validation, F.M., X.W., and D.W.; Formal Analysis, F.M. and F.S.; Investigation, F.M., X.W., and F.S.; Resources, X.W. and D.W.; Data Curation, X.W.; Writing—Original Draft Preparation, F.M., D.W., F.S., L.F., and X.W.; Writing—Review & Editing, F.M.; Visualization, F.M.; Supervision, F.M.; Project Administration, X.W.; Funding Acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (No.2016YFC0802904), National Natural Science Foundation of China (No.61671470), Natural Science Foundation of Jiangsu Province (BK20161470), 62nd batch of funded projects of China Postdoctoral Science Foundation (No. 2017M623423).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Haoze, S.; Tianqing, C.; Lei, Z.; Guozhen, Y.; Bin, H.; Junwei, C. Armored target detection in battlefield environment based on top-down aggregation network and hierarchical scale optimization. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 312–370. [CrossRef]
2. Haoze, S.; Tianqing, C.; Quandong, W.; Depeng, K.; Wenjun, D. Image detection method for tank and armored targets based on hierarchical multi-scale convolution feature extraction. *Acta Armamentarii* **2017**, *38*, 1681–1691. [CrossRef]
3. Qi, C.; Wanli, O.; Hongsheng, L.; Xiaogang, W.; Liu, B.; Yu, N. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
4. Gundogdu, E.; Alatan, A.A. Good features to correlate for visual tracking. *IEEE Trans. Image Process.* **2018**, *27*, 2526–2540. [CrossRef] [PubMed]
5. Fantacci, C.; Vo, B.N.; Vo, B.T.; Battistelli, G.; Chisci, L. Robust fusion for multisensor multiobject tracking. *IEEE Signal Process. Lett.* **2018**, *25*, 640–644. [CrossRef]
6. Jia, B.; Lv, J.; Liu, D. Deep learning-based automatic downbeat tracking: A brief review. *Multimedia Systems* **2019**, 1–22. [CrossRef]
7. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. *arXiv* **2018**, arXiv:1812.11703.
8. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H.S. Fast online object tracking and segmentation: A unifying approach. *arXiv* **2018**, arXiv:1812.05050.
9. Melekhov, I.; Kannala, J.; Rahtu, E. Siamese network features for image matching. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016. [CrossRef]
10. Yicong, T.; Afshin, D.; Mubarak, S. On detection, data association and segmentation for multi-target tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2146–2160. [CrossRef]
11. Dawei, Z.; Hao, F.; Liang, X.; Tao, W.; Bin, D. Multi-object tracking with correlation filter for autonomous vehicle. *Sensors* **2018**, *18*, 2004. [CrossRef]
12. Yang, M.; Wu, Y.; Jia, Y. A hybrid data association framework for robust online multi-object tracking. *IEEE Trans. Image Process.* **2017**, *26*, 5667–5679. [CrossRef]
13. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13–18 June 2010. [CrossRef]
14. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
15. Min, J.; Jianyu, S.; Jun, K.; Hongtao, H. Regularisation learning of correlation filters for robust visual tracking. *IET Image Process.* **2018**, *12*, 1586–1594. [CrossRef]
16. Kuai, Y.; Wen, G.; Li, D. Learning adaptively windowed correlation filters for robust tracking. *J. Visual Comm. Image Represent.* **2018**, *51*, 104–111. [CrossRef]

17. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yang, M.H. Learning spatial-temporal regularized correlation filters for visual tracking. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
18. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada, 3–6 December 2012. [CrossRef]
19. Tom, Y.; Devamanyu, H.; Soujanya, P.; Erik, C. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
20. Han, J.; Zhang, D.; Cheng, G.; Liu, N.; Xu, D. Advanced deep-learning techniques for salient and category-specific object detection: A survey. *IEEE Signal Process. Mag.* **2018**, *35*, 84–100. [CrossRef]
21. Chin, T.W.; Yu, C.L.; Halpern, M.; Genc, H.; Tsao, S.L.; Reddi, V.J. Domain-Specific Approximation for Object Detection. *IEEE Micro* **2018**, *38*, 31–40. [CrossRef]
22. Ranjan, R.; Sankaranarayanan, S.; Bansal, A.; Bodla, N.; Chen, J.C.; Patel, V.M.; Castillo, C.D.; Chellappa, R. Deep learning for understanding faces: Machines may be just as good, or better, than humans. *IEEE Signal Process. Mag.* **2018**, *35*, 66–83. [CrossRef]
23. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A. Mots: Multi-object tracking and segmentation. *arXiv* **2018**, arXiv:1902.03604.
24. Seguin, G.; Bojanowski, P.; Lajugie, R.; Laptev, I. Instance-Level Video Segmentation from Object Tracks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
25. Sadeghian, A.; Alahi, A.; Savarese, S. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
26. Babenko, B.; Yang, M.H.; Belongie, S. Visual tracking with online Multiple Instance Learning. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009. [CrossRef]
27. Smeulders, A.W.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual tracking: An experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1442–1468. [CrossRef]
28. Yang, W.; Liu, Y.; Zhang, Q.; Zheng, Y. Comparative object similarity learning-based robust visual tracking. *IEEE Access* **2019**, *7*, 50466–50475. [CrossRef]
29. Son, J.; Baek, M.; Cho, M.; Han, B. Multi-object Tracking with Quadruplet Convolutional Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [CrossRef] [PubMed]
31. Sarikaya, D.; Corso, J.; Guru, K. Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection. *IEEE Trans. Med. Imaging* **2017**, *36*, 1542–1549. [CrossRef] [PubMed]
32. Zhong, Z.; Sun, L.; Huo, Q. An anchor-free region proposal network for faster r-cnn based text detection approaches. *Int. J. Doc. Anal. Recognit.* **2019**, *22*, 315. [CrossRef]
33. Sun, X.; Wu, P.; Hoi, S.C.H. Face detection using deep learning:an improved faster rcnn approach. *Neurocomputing* **2018**, *299*, 42–50. [CrossRef]
34. Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. Domain adaptive faster r-cnn for object detection in the wild. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
35. Giuseppe, S.; Massimiliano, G.; Antonio, M.; Raffaele, G. A cnn-based fusion method for feature extraction from sentinel data. *Remote Sens.* **2018**, *10*, 236. [CrossRef]
36. Wang, J.; Chen, K.; Yang, S.; Loy, C.; Lin, D. Region proposal by guided anchoring. *arXiv* **2019**, arXiv:1901.03278.
37. Yeung, F.; Levinson, S.F.; Parker, K.J. Multilevel and motion model-based ultrasonic speckle tracking algorithms. *Ultrasound Med. Biol.* **1998**, *24*, 427–441. [CrossRef]
38. Park, J.D.; Doherty, J.F. Track detection of low observable targets using a motion model. *IEEE Access* **2015**, *3*, 1408–1415. [CrossRef]

39. Bae, S.H.; Yoon, K.J. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 595–610. [CrossRef]
40. Henschel, R.; Leal-Taixé, L.; Cremers, D.; Rosenhahn, B. Fusion of head and full-body detectors for multi-object tracking. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018. [CrossRef]
41. Long, L.; Xinchao, W.; Shiliang, Z.; Dacheng, T.; Wen, G.; Huang, T.S. Interacting tracklets for multi-object tracking. *IEEE Trans. Image Process.* **2018**, *27*, 4585–4597. [CrossRef]
42. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 318–327. [CrossRef] [PubMed]
43. Leal-Taixe, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv* **2015**, arXiv:1504.01942.
44. Milan, A.; Leal-Taixe, L.; Reid, I.; Roth, S.; Schindler, K. Mot16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
45. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012. [CrossRef]
46. Keni, B.; Rainer, S. Evaluating multiple object tracking performance: The clear mot metrics', eurasip. *EURASIP J. Image Video Proc.* **2008**, *1*, 246309. [CrossRef]
47. Leal-Taixé, L.; Ferrer, C.C.; Schindler, K. Learning by tracking: Siamese cnn for robust target association. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016. [CrossRef]
48. Wang, B.; Wang, L.; Shuai, B.; Zuo, Z.; Wang, G. Joint Learning of Convolutional Neural Networks and Temporally Constrained Metrics for Tracklet Association. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016. [CrossRef]
49. Milan, A.; Schindler, K.; Roth, S. Multi-target tracking by discrete-continuous energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2054–2068. [CrossRef]
50. Wang, S.; Fowlkes, C.C. Learning optimal parameters for multi-target tracking with contextual interactions. *Int. J. Comput. Vis.* **2016**, *122*, 1–18. [CrossRef]
51. Kieritz, H.; Becker, S.; Hubner, W.; Arens, M. Online multi-person tracking using Integral Channel Features. In Proceedings of the 2016 13th IEEE International Conference on Advanced Video and Signal BasedSurveillance (AVSS), Colorado Springs, CO, USA, 23–26 August 2016. [CrossRef]
52. Yoon, J.H.; Lee, C.R.; Yang, M.H.; Yoon, K.J. Online multi-object tracking via structural constraint event aggregation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016. [CrossRef]
53. Lenz, P.; Geiger, A.; Urtasun, R. FollowMe: Efficient Online Min-Cost Flow Tracking with Bounded Memory and Computation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) 2015, Santiago, Chile, 7–13 December 2015. [CrossRef]
54. Zhang, W.; Zhou, H.; Sun, S.; Wang, Z.; Shi, J.; Loy, C.C. Robust multi-modality multi-object tracking. Proceedings of The IEEE International Conference on Computer Vision (ICCV) 2019, Seoul, Korea, 27 October–3 November 2019.
55. Sharma, S.; Ansari, J.A.; Murthy, J.K.; Krishna, K.M. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In Proceedings of the IEEE Conference on Robotics and Automation (ICRA) 2018, Brisbane, QLD, Australia, 21–25 May 2018. [CrossRef]