Exploiting All Mamba Fusion for Efficient RGB-D Tracking

Ge Ying[*1, 2], **Dawei Zhang**[*2, 3], **Chengzhuan Yang**[2], **Wei Liu**[4], **Sang-Woon Jeon**[2], **Hua Wang**[6], **Changqin Huang**[1], **Zhonglong Zheng**[1, 2, 5†]

[1]Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Jinhua, China
[2]School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
[3]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[4]School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China
[5]China-Mozambique "Belt and Road" Joint Laboratory on Smart Agriculture, Zhejiang Normal University, Jinhua, China
[6]Institute for Sustainable Industries and Liveable Cities, College of Engineering and Science, Victoria University, Melbourne, Australia
{yingge, davidzhang, czyang}@zjnu.edu.cn, weiliucv@sjtu.edu.cn, sangwoonjeon@hanyang.ac.kr, hua.wang@vu.edu.au, cqhuang@zju.edu.cn, zhonglong@zjnu.edu.cn

Abstract

Despite the progress made through deep learning, existing Visual Object Tracking (VOT) frameworks struggle with real-world challenges. Recent approaches incorporate additional modalities like Depth, Thermal Infrared, and Language to enhance the robustness of VOT, particularly with the improvement of the depth sensor precision, facilitating RGB-D tracking. However, current RGB-D trackers often copy RGB tracking paradigms, leading to inefficiency due to two-stream architectures that fail to exploit heterogeneous features, and reliance on simplistic or large-parameter fusion methods. To address these challenges, we propose AMTrack, a one-stream RGB-D tracker leveraging Mamba's linear complexity for simultaneous feature extraction and two-stage cross-modal feature fusion. Our innovation also includes a low-parameter Multimodal Mix Mamba (3M) module, which optimizes deep feature fusion and reduces computational overhead. The advantage of the 3M module stems from our Multimodal State Space Model (MSSM), a multimodal feature interaction component reconstructed based on SSM. Experiments across multiple RGB-D tracking datasets indicate that AMTrack achieves superior performance with lower parameters and memory demands compared to state-of-the-arts.

Introduction

Visual Object Tracking (VOT) aims to predict the size and position of a target in subsequent frames of a video sequence, given the target's size and location in the initial frame. With the proliferation and advancement of deep learning technologies, VOT has become a critical downstream task in computer vision and has been widely applied in various domains, such as autonomous driving, security surveillance, and military missile guidance. Although existing VOT frameworks demonstrate robust performance in some standard scenes (Ye et al. 2022; Zheng et al. 2024), they fail to effectively track objects in real-world applications, particularly in the presence of severe occlusions, object deformation, and extreme illumination variations (Yang et al. 2022). Despite the introduction of large-scale, multi-scene, long-term and short-term tracking datasets (Fan et al. 2019; Muller et al. 2018; Mueller, Smith, and Ghanem 2016) to enrich the training data or add more temporal information (Zheng et al. 2024), these efforts still do not overcome the tracking limitations of traditional VOT. Faced with this significant challenge in the practical application of VOT, an increasing number of researchers have recognized the inherent limitations of the RGB-only data in special scenarios. They have started to address the robustness issues for RGB-only object tracking by incorporating additional modalities of image information. Depending on the type of sensor data collected, the additional modalities primarily include Depth, Thermal Infrared, Event, and Language. Some trackers target only one additional mode (Zhu et al. 2023; Hui et al. 2023; Cao et al. 2024), while others are unified multimodal

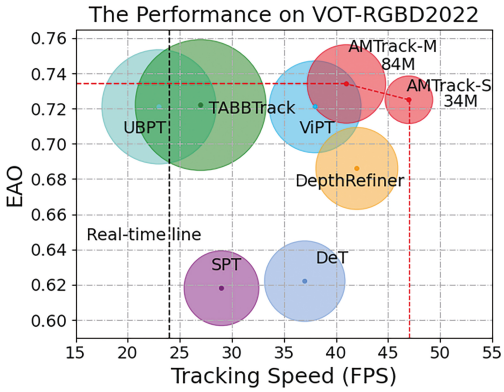

Figure 1: Comparison of AMTrack with existing trackers in terms of EAO and Speed on VOT-RGBD2022. The size of the circle represents the scale of the tracker's parameters.

[*]These authors contributed equally.
[†]Corresponding Author.

trackers (Hu et al. 2025b; Zheng et al. 2025).

In recent years, with the advancement of depth sensors, the accuracy of depth information capture has significantly improved. The enhancement in depth data quality has greatly facilitated the development of RGB-D object tracking. Recently, RGB-D trackers have been proposed using CNNs (Yan et al. 2021; Zhao et al. 2021; Kristan et al. 2019) or Transformers (Ying et al. 2025; Ye, Xiao, and Liu 2024; Ou et al. 2024) with shared weights as backbone networks, which enhance the ability to extract multimodal features. This approach also changes the classical paradigm of earlier RGB-D trackers (Camplani et al. 2015; Kart, Kmrinen, and Matas 2018) where depth information is used as an auxiliary modality for handcrafted feature tracking. Although existing RGB-D object trackers have shown slightly better performance than RGB-only trackers in certain scenarios, several unresolved issues remain: 1. Existing RGB-D object trackers (Qian et al. 2021; Camplani et al. 2015; Zhao et al. 2021) often mimic RGB trackers but struggle to balance robustness and efficiency due to the additional Depth input; 2. Current RGB-D trackers use dual-stream architectures with shared weights (Kristan et al. 2019, 2020), enhancing only homogeneous features and neglecting heterogeneous ones, conflicting with true multimodal fusion; 3. At present, RGB-D trackers use either simplistic or parameter-heavy fusion strategies (Kristan et al. 2021, 2022; Ying et al. 2025), like Transformer-based cross-attention, which fail to effectively balance robustness and efficiency in fusion.

To address the aforementioned issues in RGB-D object tracking, our approach begins by dividing the search region and template region of both RGB and Depth modalities into patches and completes feature embedding through linear projection. Inspired by the efficient state space modeling capability of the Mamba architecture and its advantage of linear complexity in processing long sequences. These preliminary features from both modalities are then input into a Mamba-based one-stream backbone network, simultaneously conducting feature extraction and the first stage of multimodal feature fusion. Considering the drawbacks brought by overly simplistic or excessively parameter-heavy fusion modules, we also design a low-parameter, easily trainable Multimodal Mix Mamba (3M) module for the second stage of deep multimodal feature fusion.

The main contributions and innovations of this paper can be summarized as follows:

- To the best of our knowledge, we are the first to utilize the linear complexity advantage of Mamba long-sequence modeling to create a one-stream RGB-D object tracking framework, performing feature extraction and modality fusion simultaneously in the same stage.
- The proposed 3M module uses the low-parameter MSSM (Multimodal State Space Model) component to reinforce the second stage of deep feature fusion between RGB and depth modalities, further enhancing the cross-modal feature fusion capability of the one-stream framework.
- Benefiting from the all Mamba framework, our AM-Track, after training on a small-scale RGB-D dataset, demonstrates its advantages on multiple public RGB-D tracking datasets against the state-of-the-art trackers with lower parameters and memory requirements.

Related Works

RGB-D Tracking

RGB-D object tracking aims to combine RGB and Depth to enhance the robustness of tracking algorithms in challenging scenarios such as severe occlusion and extreme illumination variations. In 2013, the first large-scale RGB-D dataset, the Princeton Tracking Benchmark (Song and Xiao 2013) was introduced which marked the beginning of formal research related to RGB-D tracking. Early trackers typically relied on handcrafted features for tracking, leading to significant performance variations across different scenarios and poor generalization (García et al. 2012; Camplani et al. 2015; Xiao et al. 2017). Yan et al. proposed an end-to-end offline tracker DeT (Yan et al. 2021) which incorporates depth images into the pre-training process. The excellent performance of DeT unveils that depth information is equally important as RGB information in RGB-D tracking, providing geometric and spatial location information that RGB cannot offer. Lai et al. constructed the MixForRGBD (Kristan et al. 2022) based on the Transformer framework, achieving outstanding results in the VOT-RGBD2022 challenge by improving the multimodal fusion paradigm and several fusion components using MixFormer (Cui et al. 2022). This demonstrates the great potential of the Transformer framework in multimodal tracking. Prompt learning-based trackers (Zhu et al. 2023; Ou et al. 2024) also shows well effect. Recently, some research (Tan et al. 2025; Hong et al. 2024; Wu et al. 2024) has developed unified multimodal tracking frameworks. SUTrack (Chen et al. 2025) further introduced a task-recognition auxiliary training strategy for five types of multimodal tracking. We aim to introduce a Mamba-based one-stream framework that simplifies the complex multi-stream architecture of existing RGB-D object tracking frameworks.

State Space Model and Vision Mamba

The concept of the State Space Model (SSM) was first introduced in control theory as a mathematical model for representing dynamic systems (Kalman 1960). Gu et al. proposed a novel state space framework called Mamba (Gu and Dao 2023), based on the structured SSM v6 suitable for deep learning. Zhu et al. are the first to apply the Mamba architecture to image classification (Zhu et al. 2024). To leverage Mamba's long sequence modeling capability and linear efficiency in video-based visual downstream tasks, Li et al. proposed Vim, a general visual backbone network with bidirectional Mamba blocks, demonstrating significant potential in long-term video understanding (Liu et al. 2024). In multimodal tracking, Huang et al. utilized the Mamba backbone to extract event stream information and enhance interactive learning between RGB and event streams (Huang et al. 2024; Hu et al. 2025a), while Lai et al. applied Mamba for spatio-temporal context modeling for RGB-T tracking (Lai et al. 2025). Inspired by above works, we aim to employ Mamba for a two-stage interactive fusion of RGB and Depth

modalities, optimizing the multimodal fusion paradigm and efficiency of RGB-D tracking.

Methodology

In this section, we provide a detailed introduction to the proposed AMTrack shown in Figure 2. First, we introduce the foundational knowledge related to SSM and Mamba. Next, we present the Mamba-based one-stream RGB-D backbone network and the 3M module.

SSMs and Mamba

SSM was initially introduced in Kalman filtering. After years of development, SSM has evolved into a series of simple but efficient deep sequence modeling frameworks. It maps the input sequence $x(t) \in \mathbb{R}^L$ to $y(t) \in \mathbb{R}^L$ through a hidden state $h(t) \in \mathbb{R}^N$. The process in continuous systems can be represented as:

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t) \\ y(t) &= \mathbf{C}h(t) + \mathbf{D}x(t) \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the evolution matrix, $\mathbf{B} \in \mathbb{R}^{N\times L}$ is the projection matrix, $\mathbf{C} \in \mathbb{R}^{L\times N}$ is the output matrix, and $\mathbf{D} \in \mathbb{R}^{L\times L}$ is the feedthrough matrix provided direct signals from input to output, similar to skip connections. $h'(t) \in \mathbb{R}^N$ is the derivative of the hidden state.

To handle discrete inputs such as text or video sequences, SSM uses zero-order hold techniques to convert discrete data into a continuous form, the derivation process is:

$$\begin{aligned} h_t &= \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t, \\ y_t &= \overline{\mathbf{C}}h_t \end{aligned} \tag{2}$$

where h_t and h_{t-1} represent the discrete hidden states, x_t and y_t represent the discrete inputs and outputs. $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$ and $\overline{\mathbf{C}}$ are discrete matrix of SSM, according to the learnable step Δ, the representation formula of the SSM utilizing a selective scanning mechanism (S6) for discrete systems is shown as follows:

$$\begin{aligned} \overline{\mathbf{A}} &= \exp(\Delta\mathbf{A}) \\ \overline{\mathbf{B}} &= (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B} \\ \overline{\mathbf{C}} &= \mathbf{C} \end{aligned} \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{D\times N}$, $\mathbf{B} \in \mathbb{R}^{B\times L\times N}$, $\mathbf{C} \in \mathbb{R}^{B\times L\times N}$ and $\Delta \in \mathbb{R}^{B\times L\times D}$. The parameters are produced by linear projection based on the input $x_t \in \mathbb{R}^{B\times L\times D}$.

Bidirectional Mamba Backbone

To facilitate subsequent alignment of multimodal information, we first map the Depth image to a pseudo-color map form of dimension 3, following the RGB-D tracking convention. The RGB images and Depth images are both adopted as the input to our Bidirectional Mamba Backbone. RGB and Depth template images are denoted as $Z_{rgb} \in \mathbb{R}^{H_z\times W_z\times 3}$ and $Z_{dep} \in \mathbb{R}^{H_z\times W_z\times 3}$, while RGB and Depth search images are denoted as $X_{rgb} \in \mathbb{R}^{H_x\times W_x\times 3}$ and $X_{dep} \in \mathbb{R}^{H_x\times W_x\times 3}$. As illustrated in Figure 2, four input images are first projected into tokens and then feed into a $P \times P$ sized two-dimensional convolution with equal steps, and the final convolution result is flattened into one-dimensional tokens, denoted as $X'_{rgb} \in \mathbb{R}^{N_x\times D}$ ($D = 3P^2$), $X'_{dep} \in \mathbb{R}^{N_x\times D}$, $Z'_{rgb} \in \mathbb{R}^{N_z\times D}$, and $Z'_{dep} \in \mathbb{R}^{N_z\times D}$, where $N_x = H_x \times W_x/P^2$ and $N_z = H_z \times W_z/P^2$ symbolize the token number of the template and search image inputs. Then we add position embedding to provide location prior information. In order to build a one-stream backbone, we combine the four groups of token embedding into E :

$$\begin{aligned} E^0 &= [Z'_{rgb} + P_z; X'_{rgb} + P_x; Z'_{dep} + P_z; X'_{dep} + P_x] \\ E^l &= Vim(E^{l-1}) + E^{l-1}, \quad l = 1, 2, ..., L \end{aligned} \tag{4}$$

where $P_x \in \mathbb{R}^{N_x\times D}$ and $P_z \in \mathbb{R}^{N_z\times D}$, E^0 denotes initial features of merging and E^{l-1} denotes (l-1)-th layer features.

The detailed process of the bidirectional mamba block is outlined as follows, to obtain the features of the l-th layer, we first pass the features of the (l-1)-th layer through a normalization layer. The normalized features are then passed through two linear projection layers to obtain two preprocessed feature vectors, x and z, which are used as inputs for the subsequent SSM module.

$$\begin{aligned} x &= (Linear^x(Norm(E^{l-1}))) \\ z &= (Linear^z(Norm(E^{l-1}))) \end{aligned} \tag{5}$$

We process the vector x in both the forward and backward directions, using a bidirectional mode. This is because the pixel-by-pixel relationship in an image preserves a two-dimensional spatial relationship, unlike the one-dimensional forward-backward relationship between word phrases in the NLP community. The bidirectional sampling mode does not make the feature processing overly complex, nor does it excessively lose the two-dimensional spatial relationship between individual pixels. Then, based on the feature x', we perform multiple linear projections to obtain the key input matrix $\mathbf{B}$, the output matrix $\mathbf{C}$, and the time step Δ for this discrete system. The entire process can be represented as:

$$x' = SiLU(Conv1d(x)) \tag{6}$$

$$\mathbf{B}, \mathbf{C}, \Delta = Linear(x'), \quad \overline{\mathbf{A}}, \overline{\mathbf{B}} = ZOH(\mathbf{A}, \mathbf{B}, \Delta) \tag{7}$$

where $Conv1d$ denotes the depth-wise convolution, $SiLU$ denotes SiLU activation function, ZOH denotes the zero-order hold method. Finally, we multiply and integrate the forward and backward results with the residuals z which have the same skip connection properties to obtain y':

$$\begin{aligned} y' = SSM_{forward}(x') \odot SiLU(z) + \\ SSM_{backward}(x') \odot SiLU(z) \end{aligned} \tag{8}$$

Specifically, the calculation process of forward or backward SSM can be both expressed as:

$$h' = \overline{\mathbf{A}}h + \overline{\mathbf{B}}x', \quad y = \mathbf{C}h' \tag{9}$$

where h' denotes the hidden state matrix of this layer, obtained from the hidden matrix h of the previous layer. At this

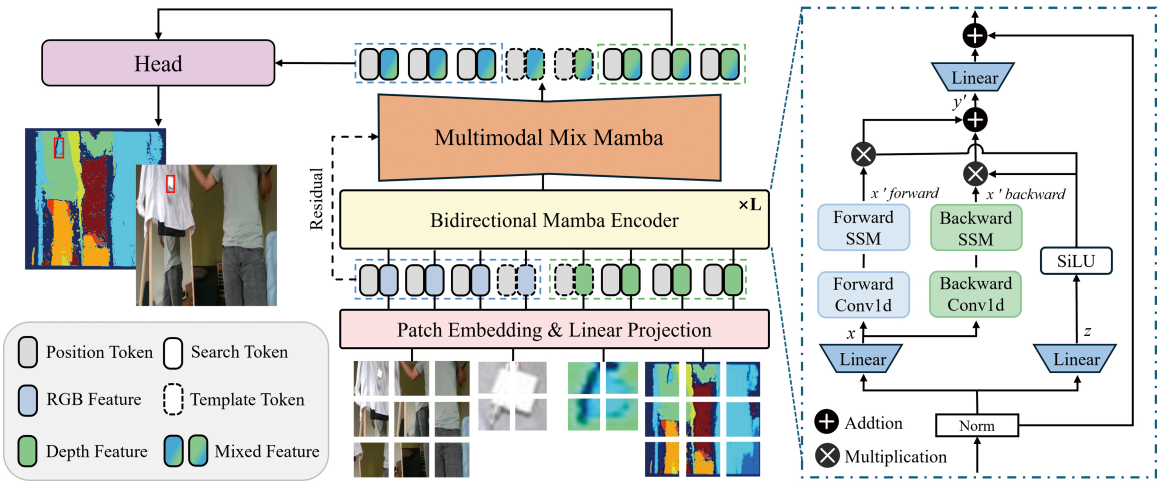

Figure 2: The overall architecture of our proposed AMTrack. RGB and Depth images are embedded as tokens and sent together to the Bidirectional Mamba Encoder with L layers for the first stage of feature extraction and interaction, and then enter the 3M module for the second stage of deep feature cross-modal interaction.

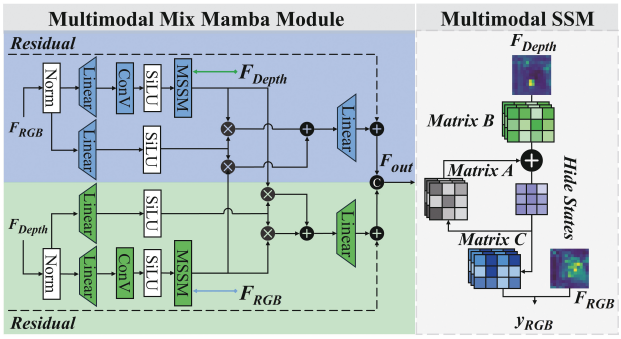

Figure 3: Details of our proposed 3M module. Left shows how to perform cross-modality feature interaction in the second stage, right side is the core component MSSM based on SSM reconstruction.

point, we construct a one-stream backbone network based on Mamba, which performs feature extraction and the first stage of cross-modal fusion on four groups of images from two modalities simultaneously, significantly reducing the redundancy of the RGB-D tracking framework.

Multimodal Mix Mamba Module

Previous RGB-D trackers often use cross-modal fusion modules which are redundant and miscellaneous. These modules are designed to perform saturated cross-modal interactions after extracting single-modal features. This saturated interaction paradigm, while enhancing homogeneous features, also amplifies the inherent noise of each modality, thereby affecting tracking robustness. Inspired by specific mamba-form fusion networks such as FusionMamba (Peng et al. 2024) and Mamba-FETrack (Huang et al. 2024), we restructure the components of the mamba module to achieve deep cross-modal fusion of RGB and Depth modalities in the second stage. The components of the mamba module maintain the original linear complexity and long-range dependency characteristics, enabling the 3M module to possess cross-modal feature fusion while having fewer parameters and lower computational cost than traditional cross-modal fusion modules. The non-saturated cross-modal interaction paradigm also avoids the drawback of different modal noises amplifying each other, which is often present in traditional fusion modules. Details are shown in Figure 3.

To more clearly articulate the computation process of the 3M module, we describe it below by using the RGB branch as the main input to the 3M module and the Depth branch as the auxiliary input, the calculation method for the other branch is the same. We first normalize the features of the RGB image after passing through the l-layers of the Mamba backbone network. Then, the normalized features are processed by a designated linear layer to generate two independent features. After passing through a convolution and an activation function, the first independent feature is used as the main input to the MSSM component. The second independent feature is retained as the initial feature of the RGB image and is used for self-modal completion with RGB and cross-modal complementation with Depth. These operations can be formulated as:

$$\begin{aligned} F'_{RGB} &= SiLU(Conv1(Linear(Norm(F_{RGB})))) \\ F''_{RGB} &= SiLU(Linear(Norm(F_{RGB}))) \end{aligned} \tag{10}$$

where F_{RGB} denotes the RGB features extracted by the l-layers Mamba backbone, F'_{RGB}, F''_{RGB} denote the first independent feature as the main input to the MSSM and the second independent feature retained as the initial feature.

After obtaining four independent features for the respective modalities, taking the RGB branch as an example, we use F'_{RGB} as the primary input of the MSSM and F_{Depth} as the auxiliary input for cross-modal fusion of depth features in the second stage. Then, the interacted intermediate result is multiplied with features from the RGB modality

and the Depth modality separately and summed to obtain the second-stage fusion output F_{out}^{RGB} for the RGB branch. These operations can be expressed as:

$$\begin{aligned} F_{out}^{RGB} = \quad & MSSM(F'_{RGB}, F_{Depth}) \odot F''_{RGB} \quad + \\ & MSSM(F'_{Depth}, F_{RGB}) \odot F''_{RGB} \end{aligned} \tag{11}$$

The MSSM we proposed is a multimodal variant of SSM v6, featuring multiple inputs and a single output value. The two inputs for MSSM come from the deep features of the RGB modality and the Depth modality. We associate the matrix $\mathbf{B}$, affecting the input x, and the matrix $\mathbf{C}$, affecting the hidden state h, with the input of the auxiliary modality. The calculation of the final output y, however, is associated with the primary modality F_{RGB}. This configuration of matrices $\mathbf{B}$ and $\mathbf{C}$ and the computation of result y allow the MSSM to learn features of the auxiliary modality, enabling more precise control over how the primary modality input updates the hidden state h or how the hidden state h impacts the output y during the SSM computation. The algorithmic process of our MSSM is shown in Algorithm 1.

Finally, we take the multimodal features computed in the aforementioned manner and add the original residual features from their respective modalities. Then, we pass them through a linear layer and concatenate them together. The final output features of the second stage cross-modal fusion consist of four parts: the search region and template region from both RGB modality and Depth modality, the calculation formula is as follows:

$$\begin{aligned} F_{out} = \quad & Concat(Linear(F_{out}^{RGB} + R_{RGB}), \\ & Linear(F_{out}^{Depth} + R_{Depth})) \end{aligned} \tag{12}$$

where F_{out}^{RGB} and F_{out}^{Depth} denote the feature outputs when the RGB modality and Depth modality are respectively interacted as the primary modality, while R_{RGB} and R_{Depth} represent the residuals from different modalities.

Finally, we combine the final search region features of the two modalities through several linear layers, and feed them into the designed tracking head to obtain the prediction results. Following the common methods in SOT, the tracking head includes three convolutional layers to predict the target classification score map, the local offsets, and the normalized bounding box, respectively.

Algorithm 1: Multimodal State Space Model

Input: F_{RGB}, F_{Depth}: (B, L, C)
/* Respective modal features after first stage fusion */
Output: y_{RGB}: (B, L, C)
/* RGB features after second stage fusion */
$\mathbf{A}$: $(C, N) \leftarrow \text{Parameter}_{\mathbf{A}}$
$\mathbf{B}$: $(B, L, N) \leftarrow \text{Linear}_{\mathbf{B}}(F_{Depth})$
$\mathbf{C}$: $(B, L, N) \leftarrow \text{Linear}_{\mathbf{C}}(F_{Depth})$
Δ: $(B, L, C) \leftarrow \log(1 + \exp(\text{Linear}_{\Delta}(F_{Depth}) + \text{Parameter}_{\Delta}))$
$\overline{\mathbf{A}}$: $(B, L, C, N) \leftarrow \exp(\Delta \otimes \mathbf{A})$
$\overline{\mathbf{B}}$: $(B, L, C, N) \leftarrow \Delta \otimes \mathbf{B}$
$y_{RGB} \leftarrow \text{SSM}(\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{C})(F_{RGB})$
return y_{RGB}

Experiments

Implementation Details

AMTrack is implemented in Python3.9 using PyTorch2.1 (Paszke et al. 2019). The models are trained on two NVIDIA RTX3090 GPUs and the inference is performed on a single NVIDIA RTX3090 GPU. We select only the DepthTrack dataset as the training set. During the training phase, the input sizes of the template region and the search region images are adjusted to 128×128 and 256×256 respectively. Our model is trained for 20 epochs, with each epoch consisting of 60,000 image pairs and the mini-batch size of 32 sample pairs. We train our model utilizing the AdamW (Loshchilov and Hutter 2017) optimizer with weight decay set to 1×10^{-4}, an initial learning rate of 1×10^{-4}, which is reduced by a factor of 1×10^{-5} after 12 epochs.

Model Variants. To further validate the effectiveness of the model, we train models of different sizes as follows:

- **AMTrack-S.** Backbone size [Vim-Small 384]
- **AMTrack-M.** Backbone size [Vim-Middle 576]

Comparison with SOTA Trackers

To highlight the advantages of AMTrack, we compare it with numerous SOTA tracking algorithms in Table 1, where we carefully distinguish between different types of algorithms. "RGB-X" indicates that it is a unified multimodal tracking framework, such as ViPT, Un-Track.

AMTrack-M achieves an F-score of 74.2 on the CDTB dataset. From multiple experiments, we find that the poor performance of the depth sensors used to capture the CDTB dataset leads to more noise points in distant backgrounds, reducing the quality of depth images, which in turn affects the MSSM component to some extent. Therefore, in the CDTB dataset, AMTrack does not create a significant gap compared to Transformer-based algorithms. On the DepthTrack dataset, AMTrack-M achieves an F-score of 63.4, outperforming unified multimodal tracking frameworks such as OneTracker (Hong et al. 2024) and Un-Track (Wu et al. 2024). This demonstrates that AMTrack, designed specifically for RGB-D target tracking scenarios, can achieve better tracking performance when using downstream modality training sets of the same scale. In the more complex RGB-D tracking scenarios of the ARKitTrack test set, both AMTrack-S and AMTrack-M obtain the leading results, indicating that the tracking framework proposed in this paper can maintain good robustness in complex scenarios.

To further demonstrate that the all Mamba architecture of AMTrack possesses strong generalization capabilities while maintaining a minimal number of parameters, we conduct a comparison of parameters, speed, and EAO among several representative algorithms on VOT-RGBD2022 provided by the VOT-Challenge. The results are presented in Table 2. Evidently, AMTrack-S achieves an F-score of 72.5 with only 34M parameters. Compared to the CNN-based architecture of DeT and Transformer-based models, AMTrack uses Mamba as the paradigm for its one-stream backbone network, significantly enhancing inference speed and ensuring the effectiveness of multimodal feature extraction and fusion.

Method	Year	CDTB			DepthTrack			ARKitTrack			Backbone	Publication	Type
		Pr↑	Re↑	F-score↑	Pr↑	Re↑	F-score↑	Pr↑	Re↑	F-score↑			
STARK-ST101	2021	65.7	66.9	66.3	50.3	46.8	48.5	40.7	38.1	39.3	ResNet-101	ICCV	RGB
OSTrack	2022	71.3	68.6	69.9	57.2	56.3	56.7	44.0	44.0	44.0	ViT-B	ECCV	RGB
MixFormer1k	2022	69.2	66.4	67.8	49.0	45.4	47.1	44.9	42.1	43.4	CvT	CVPR	RGB
ToMP101	2022	67.0	68.3	67.6	51.5	49.5	50.5	44.9	43.3	44.1	ResNet-101	CVPR	RGB
ATCAIS	2020	70.9	69.6	70.2	47.3	40.2	43.5	38.9	34.3	36.4	ResNet-18	VOT-2020	RGB-D
DAL	2020	62.0	56.0	58.9	51.2	36.9	42.9	44.6	32.9	37.8	ResNet-18	ICPR	RGB-D
DDiMP	2020	70.3	68.9	69.6	54.0	47.5	50.6	49.5	41.3	45.0	ResNet-50	VOT-2020	RGB-D
TSDM	2021	64.7	54.3	59.1	44.2	36.3	39.8	38.9	29.2	33.4	ResNet-50	ICPR	RGB-D
DeT	2021	67.4	64.2	65.7	56.0	50.6	53.2	42.8	40.5	41.6	ResNet-50	ICCV	RGB-D
TALGD	2022	63.0	59.6	61.3	49.4	42.4	45.6	42.8	35.2	38.6	ResNet-50	VOT-2021	RGB-D
ProTrack	2022	-	-	-	57.8	57.3	58.3	-	-	-	-	ACM MM	RGB-X
SPT	2023	65.4	72.6	68.8	52.7	54.9	53.8	-	-	-	ResNet-50	AAAI	RGB-D
ARKitTrack	2023	71.1	67.1	69.0	61.7	60.7	61.2	48.8	46.9	47.8	ViT-B	CVPR	RGB-D
ViPT	2023	-	-	-	59.2	59.6	59.4	-	-	-	ViT-B	CVPR	RGB-X
SSLTrack	2024	65.0	62.0	63.5	56.5	49.1	52.5	-	-	-	-	PR	RGB-D
OneTracker	2024	-	-	-	60.7	60.4	60.9	-	-	-	ViT-B	CVPR	RGB-X
Un-Track	2024	-	-	-	61.3	61.0	61.2	-	-	-	ViT-B	CVPR	RGB-X
TABBTrack	2024	72.1	72.2	72.1	62.2	61.5	61.8	51.0	47.8	49.3	ViT-B	PR	RGB-D
XTrack-B	2025	-	-	-	61.8	62.0	61.5	-	-	-	MoE	ICCV	RGB-X
APTrack	2025	-	-	-	62.3	61.9	62.1	-	-	-	ViT-B	TAI	RGB-X
AMTrack-S	2025	72.3	74.1	73.2	62.1	61.2	61.7	50.9	47.9	49.4	Vim-S	-	RGB-D
AMTrack-M	2025	72.9	75.6	74.2	64.1	62.8	63.4	52.3	48.6	50.4	Vim-M	-	RGB-D

Table 1: Comparison of overall performance of AMTrack and numerous trackers on three prevalent RGB-D tracking evaluation benchmarks. Red/Green/Blue indicates the best/runner-up/third best results.

Method	Training Sets	Params	EAO	FPS
DeT	DepthTrack	69M	62.2	37
ViPT	DepthTrack	93M	72.1	38
UBPT	DepthTrack	121M	72.1	23
TABBTrack	DepthTrack	185M	72.2	27
DepthRefiner	DepthTrack	72M	68.6	42
SDSTrack	DepthTrack	108M	72.8	21
OneTracker	DepthTrack LasHeR VisEvent DAVIS16	100M	72.7	-
Un-Track	DepthTrack LasHeR VisEvent	99M	72.1	-
AMTrack-S	DepthTrack	**34M**	72.5	**47**
AMTrack-M	DepthTrack	84M	**73.4**	41

Table 2: Comparison of AMTrack for Parameters, EAO, and Inference Speed on the VOT-RGBD2022 Dataset. The results are tested on a single NVIDIA RTX3090 GPU.

Ablation Study

To thoroughly understand which components in AMTrack objectively enhance performance for RGB-D tracking, we design multiple intermediate models for comparative analysis of the benefits provided by each component. The performance evaluation in the comparison is conducted on the most widely used DepthTrack (Yan et al. 2021) dataset and results are shown in Table 3. "→" indicates the replaced components, "Δ" represents the performance changes after component replacement, and we select OSTrack (Ye et al. 2022) with dual branches as the baseline.

One-stream Framework. We compare the performance differences between the one-stream framework and the two-stream framework, with the results shown as **#3** in Table 3. The results on DepthTrack indicate that using a two-stream Vim with shared weights based on each modality as the backbone network leads to a decrease of 0.6 points in the final F-score. This demonstrates that the one-stream Vim framework achieves the first stage of shallow multimodal feature fusion more effectively during the feature extraction phase. Furthermore, the feature extraction mode with non-shared weights can more effectively filter the heterogeneous features of different modalities. To avoid the influence of the backbone network type on the one-stream and two-stream architectures that could affect the final results, as shown in **#2**, we also conduct experiments replacing Vim entirely with ViT (Dosovitskiy et al. 2020). The results demonstrate that the type of backbone network does not significantly impact the performance of the one-stream and two-stream architectures. The primary reason for the model's performance improvement is still the use of the one-stream architecture.

Bidirectional Mamba. With the rapid development of the Mamba architecture, there are now various variations of Mamba available for use in visual downstream tasks. To determine whether different Mamba structures affect the tracking performance of AMTrack, we conduct ablation experiments comparing Unidirectional and Bidirectional Mamba

#	Method	F-score	Δ
1	Baseline	56.7	-5.0
2	Vim → ViT	61.9	+0.2
3	One-stream → Two-stream	61.1	-0.6
4	Bidirectional → Undirectional	61.5	-0.2
5	W/o 3M Module	59.7	-2.0
6	MSSM → SSM	60.6	-1.1
7	**AMTrack-S**	**61.7**	-

Table 3: Ablation study on DepthTrack. Δ denotes the performance changes compared with AMTrack-S.

structures, as shown in **#4**. The results of experiment **#4** indicate that changing the unidirectional Mamba to a bidirectional structure leads to a slight increase of 0.2 in the final F-score. This suggests that the Mamba structure with a more efficient scan mode can indeed enhance feature extraction capabilities. It is evident that utilizing Quad-directional Mamba, Tree Topology Mamba (Xiao et al. 2024), etc., can certainly improve tracking performance. However, to maintain an optimal balance between robustness and speed of AMTrack, we ultimately choose to adopt only the Bidirectional Mamba structure as our component.

3M Module. To demonstrate the advantages of proposed 3M module, we remove the 3M module entirely, leaving only the first stage of feature extraction and shallow feature fusion. The results shown as **#5** in Table 3 indicate that after removing the 3M module, AMTrack's performance on DepthTrack drops by 2.0 points. This suggests that feature fusion in the first stage alone is insufficient, and the second stage of deep cross-modal feature fusion in AMTrack is essential. Our proposed 3M module, constructed using Mamba components, has a strong capability for cross-modal feature fusion, and it is also lightweight in terms of parameters and easy to train. To more intuitively demonstrate the advantages of the 3M module, we visualize a comparative analysis across multiple sets of video sequences, focusing on whether features are processed by the second-stage cross-modal interaction of the 3M module. In Figure 4, it is evident that the 3M module exhibits superior discriminative abilities in challenging scenarios, such as multiple similar objects and the reappearance of the target.

MSSM. In proposed 3M module, we redesign the traditional SSM module to have multiple inputs, allowing the auxiliary modality to guide the output of the main modality. To verify the effectiveness of proposed MSSM component, we retain the overall AMTrack framework and only replace the MSSM in the 3M module with SSM. The experimental results shown as **#6** indicate that the F-score decreased by 1.1 points. This indirectly verifies that MSSM has a gating mechanism similar to RNNs, which can suppress useless modality features during cross-modal feature interaction, thereby enhancing feature representation among different modalities for RGB-D tracking.

Multimodal Generality. To validate the universality of the proposed method across different modalities, we also

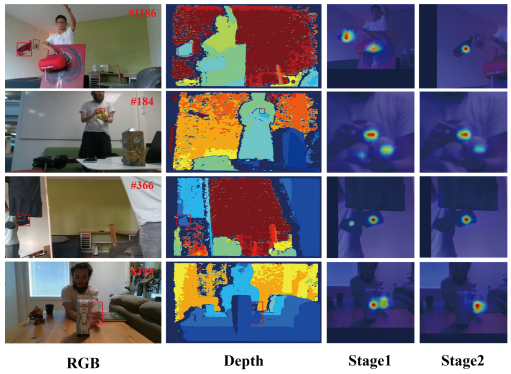

Figure 4: Visualization of the score maps in extremely challenging sequences. Stage1 denotes the score map generated solely through Bidirectional Mamba Encoder. Stage2 denotes the score map generated by the full AMTrack framework. Ground truth is annotated with red boxes.

Method	Source	RGBT234		VisEvent	
		MPR↑	MSR↑	AUC↑	Pr↑
ViPT	CVPR23	83.5	61.7	59.2	75.8
Un-Track	CVPR24	84.2	62.5	58.9	75.5
OneTracker	CVPR24	85.7	64.2	60.8	76.7
SeqTrackv2-B	ArXiv24	88.0	64.7	61.2	78.2
SUTrack-T224	AAAI25	85.9	63.8	58.8	75.7
SUTrack-B224	AAAI25	92.2	69.5	62.7	79.9
XTrack-B	ICCV25	87.4	64.9	60.9	77.5
AMTrack-M	-	**89.4**	**66.2**	**62.5**	**79.7**

Table 4: Universal performance evaluation of our AMTrack on different modality tracking datasets.

conducted additional performance evaluation on RGB-T and RGB-E tracking datasets. In Table 4, we compare the recent RGB-X trackers of the unified framework, and the results indicate that our method demonstrates good generalization capabilities for other multimodal tracking.

Conclusion

In this work, we propose an all Mamba one-stream and two-stage RGB-D tracking framework. In the first stage, Vim is employed as the backbone network to extract initial multimodal features while performing shallow cross-modal interaction. In the second stage, the reconstructed 3M module on top of Mamba achieves deep cross-modal feature fusion. The core component of the 3M module, MSSM, is specifically designed for RGB and Depth feature interaction, aiming for better homogeneous feature enhancement and heterogeneous feature decoupling. Extensive experiments on several RGB-D tracking datasets demonstrate its effectiveness. Note that the low-parameter property of AMTrack makes it easy to train. We expect the AMTrack becoming a potential RGB-D tracking solution for real-world deployment.

Acknowledgments

This work was supported in part by the Project of China-Mozambique "Belt and Road" Joint Laboratory on Smart Agriculture under Grant 2024YFE0214000, in part by the Major Program of the Natural Science Foundation of Zhejiang Province, China under Grant LD26F020003, also in part by the National Natural Science Foundation of China under Grant 62272419 and Grant 62402449, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ23F020010, and the Open Project Program of the State Key Laboratory of CAD&CG under Grant A2421.

References

Camplani, M.; Hannuna, S. L.; Mirmehdi, M.; Damen, D.; Paiement, A.; Tao, L.; and Burghardt, T. 2015. Real-time RGB-D tracking with Depth scaling kernelised correlation filters and occlusion handling. In *British Machine Vision Conference*, volume 3, 1–12.

Cao, B.; Guo, J.; Zhu, P.; and Hu, Q. 2024. Bi-directional adapter for multimodal tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 927–935.

Chen, X.; Kang, B.; Geng, W.; Zhu, J.; Liu, Y.; Wang, D.; and Lu, H. 2025. SUTrack: Towards simple and unified single object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2239–2247.

Cui, Y.; Jiang, C.; Wang, L.; and Wu, G. 2022. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13608–13618.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 1–21.

Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2019. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5374–5383.

García, G. M.; Klein, D. A.; Stückler, J.; Frintrop, S.; and Cremers, A. B. 2012. Adaptive multi-cue 3D tracking of arbitrary objects. In *Pattern Recognition: Joint 34th DAGM and 36th OAGM Symposium, Graz, Austria, August 28-31, 2012. Proceedings 34*, 357–366. Springer.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Hong, L.; Yan, S.; Zhang, R.; Li, W.; Zhou, X.; Guo, P.; Jiang, K.; Chen, Y.; Li, J.; Chen, Z.; et al. 2024. Onetracker: Unifying visual object tracking with foundation models and efficient tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19079–19091.

Hu, X.; Tai, Y.; Zhao, X.; Zhao, C.; Zhang, Z.; Li, J.; Zhong, B.; and Yang, J. 2025a. Exploiting multimodal spatial-temporal patterns for video object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 3581–3589.

Hu, X.; Zhong, B.; Liang, Q.; Shi, L.; Mo, Z.; Tai, Y.; and Yang, J. 2025b. Adaptive perception for unified visual multimodal object tracking. *IEEE Transactions on Artificial Intelligence*.

Huang, J.; Wang, S.; Wang, S.; Wu, Z.; Wang, X.; and Jiang, B. 2024. Mamba-fetrack: Frame-event tracking via state space model. In *Chinese Conference on Pattern Recognition and Computer Vision*, volume 15042, 3–18. Springer.

Hui, T.; Xun, Z.; Peng, F.; Huang, J.; Wei, X.; Wei, X.; Dai, J.; Han, J.; and Liu, S. 2023. Bridging search region interaction with template for rgb-t tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13630–13639.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82D: 35–45.

Kart, U.; Kmrinen, J. K.; and Matas, J. 2018. How to make an RGBD tracker? In *European Conference on Computer Vision 2018 Workshops*, 148–161.

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.-K.; Chang, H. J.; Danelljan, M.; Zajc, L. Č.; Lukežič, A.; et al. 2022. The tenth visual object tracking vot2022 challenge results. In *European Conference on Computer Vision 2022 Workshops*, 431–460.

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.-K.; Danelljan, M.; Zajc, L. Č.; Lukežič, A.; Drbohlav, O.; et al. 2020. The eighth visual object tracking VOT2020 challenge results. In *European Conference on Computer Vision 2020 Workshops*, 547–601.

Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.-K.; Chang, H. J.; Danelljan, M.; Cehovin, L.; Lukežič, A.; et al. 2021. The ninth visual object tracking vot2021 challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2711–2738.

Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Pflugfelder, R.; Kamarainen, J.-K.; ˇCehovin Zajc, L.; Drbohlav, O.; Lukezic, A.; Berg, A.; et al. 2019. The seventh visual object tracking vot2019 challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision workshops*, 2206–2241.

Lai, S.; Liu, C.; Zhu, J.; Kang, B.; Liu, Y.; Wang, D.; and Lu, H. 2025. Mambavt: Spatio-temporal contextual modeling for robust rgb-t tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 35: 9312–9323.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; Jiao, J.; and Liu, Y. 2024. Vmamba: Visual state space model. *Advances in Neural Information Processing Systems*, 37: 103031–103063.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Mueller, M.; Smith, N.; and Ghanem, B. 2016. A benchmark and simulator for UAV tracking. In *Proceedings of the European Conference on Computer Vision*, 445–461. Springer.

Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision*, 300–317. Springer.

Ou, Z.; Zhang, D.; Ying, G.; and Zheng, Z. 2024. UBPT: Uni-directional and Bi-directional prompts for RGBD tracking. *IEEE Sensors Journal*, 24: 37503–37513.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32: 8024–8035.

Peng, S.; Zhu, X.; Deng, H.; Deng, L.-J.; and Lei, Z. 2024. FusionMamba: Efficient remote sensing image fusion with state space model. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–16.

Qian, Y.; Yan, S.; Lukežič, A.; Kristan, M.; Kämäräinen, J.-K.; and Matas, J. 2021. DAL: A deep depth-aware long-term tracker. In *2020 25th International Conference on Pattern Recognition*, 7825–7832. IEEE.

Song, S.; and Xiao, J. 2013. Tracking revisited using RGBD camera: Unified benchmark and baselines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 233–240.

Tan, Y.; Wu, Z.; Fu, Y.; Zhou, Z.; Sun, G.; Ma, C.; Paudel, D. P.; Van Gool, L.; and Timofte, R. 2025. XTrack: Multimodal training boosts RGB-X video object trackers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5734–5744.

Wu, Z.; Zheng, J.; Ren, X.; Vasluianu, F.-A.; Ma, C.; Paudel, D. P.; Van Gool, L.; and Timofte, R. 2024. Single-model and any-modality for video object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19156–19166.

Xiao, J.; Stolkin, R.; Gao, Y.; and Leonardis, A. 2017. Robust fusion of color and depth data for RGB-D target tracking using adaptive range-invariant depth models and spatio-temporal consistency constraints. *IEEE Transactions on Cybernetics*, 48(8): 2485–2499.

Xiao, Y.; Song, L.; Huang, S.; Wang, J.; Song, S.; Ge, Y.; Li, X.; and Shan, Y. 2024. MambaTree: Tree topology is all you need in state space model. *Advances in Neural Information Processing Systems*, 37: 75329–75354.

Yan, S.; Yang, J.; Käpylä, J.; Zheng, F.; Leonardis, A.; and Kämäräinen, J.-K. 2021. Depthtrack: Unveiling the power of rgbd tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10725–10733.

Yang, J.; Li, Z.; Yan, S.; Zheng, F.; Leonardis, A.; Kämäräinen, J.-K.; and Shao, L. 2022. Rgbd object tracking: An in-depth review. *arXiv preprint arXiv:2203.14134*.

Ye, B.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2022. Joint feature learning and relation modeling for tracking: A one-stream framework. In *Proceedings of the European Conference on Computer Vision*, 341–357. Springer.

Ye, P.; Xiao, G.; and Liu, J. 2024. AMATrack: A unified network with asymmetric multimodal mixed attention for RGBD tracking. *IEEE Transactions on Instrumentation and Measurement*, 73: 1–11.

Ying, G.; Zhang, D.; Ou, Z.; Wang, X.; and Zheng, Z. 2025. Temporal adaptive bidirectional bridging for RGB-D tracking. *Pattern Recognition*, 158: 111053.

Zhao, P.; Liu, Q.; Wang, W.; and Guo, Q. 2021. Tsdm: Tracking by siamrpn++ with a depth-refiner and a mask-generator. In *2020 25th International Conference on Pattern Recognition*, 670–676. IEEE.

Zheng, Y.; Zhong, B.; Liang, Q.; Mo, Z.; Zhang, S.; and Li, X. 2024. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 7588–7596.

Zheng, Y.; Zhong, B.; Liang, Q.; Zhang, S.; Li, G.; Li, X.; and Ji, R. 2025. Towards universal modal tracking with online dense temporal token learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47: 10192–10209.

Zhu, J.; Lai, S.; Chen, X.; Wang, D.; and Lu, H. 2023. Visual prompt multi-modal tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9516–9526.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *Proceedings of the Forty-first International Conference on Machine Learning*, 62429–62442.