Competition 3: Vision + Geometry + Semantics: Collaborative Video Object Tracking with Multi-Source Heterogeneous Data

1. Competition Background

With the rapid development of artificial intelligence and computer vision technologies, visual object tracking, as one of the core technologies in intelligent perception, has significant applications in scenarios such as autonomous driving, intelligent security, human-computer interaction, and drone-based search and rescue. However, object tracking in real-world complex scenarios still faces many challenges, such as occlusion, illumination variations, background interference, and fast motion. Single-modal data (e.g., visible-light RGB) is easily affected by environmental noise, leading to degraded tracking accuracy. The advancement of multi-sensor perception technologies provides a new solution to this problem. Depth data can provide spatial geometric information of the target, enhancing robustness against scale variations and occlusion, while textual descriptions can introduce semantic priors to assist in precise localization of target identity and attributes. How to efficiently fuse the complementary advantages of multi-source heterogeneous data and overcome the bottlenecks of tracking in complex scenarios has become a common focus in both academia and industry.

Based on this background and demand, this competition aims to promote the development of accurate and robust visual object tracking algorithms in complex scenarios. Participants are encouraged to improve visual object tracking performance through innovations in multi-source data fusion and theoretical methods. The competition will provide a dataset containing visible-light RGB image sequences, depth image sequences, and textual data for object tracking. Participating teams are required to design visual object tracking algorithms that take tri-modal data as input, fully leveraging the complementary information from visual, geometric, and semantic modalities to achieve precise tracking in complex scenarios. This competition primarily evaluates participants' abilities in problem analysis, data processing, deep network construction, algorithm design and programming, as well as data visualization.

2. Competition Application Scenarios

Multi-source heterogeneous data collaboration in video object tracking has significant application value in various real-world scenarios such as autonomous driving, intelligent security, and human-computer interaction. For instance, in an intelligent security scenario, when the police need to track a suspect carrying

dangerous items in a crowded subway station, the system must fuse data from RGB cameras (capturing information such as clothing color and appearance), depth sensors (locating spatial positions), and textual cues (e.g., semantic descriptions like “a person holding a black suitcase”) to lock onto the target in real time. When the suspect suddenly crouches to hide the item, the RGB images may be partially obstructed due to viewpoint occlusion, while depth data can still track the target through geometric contours, and textual semantics can be used to validate abnormal behaviors, ultimately enabling precise interception. This scenario vividly demonstrates the core role of multi-source data collaboration in addressing occlusion, deformation, and interference, which is highly aligned with the competition’s objective of deep integration of “visual + geometric + semantic” information.

3. Competition Task

This competition aims to leverage multi-source data fusion techniques and visual object tracking technologies to achieve accurate and stable visual object tracking in complex scenarios. Specifically, it involves processing and analyzing tri-modal data inputs to enable continuous and precise tracking of the given target.

3.1 Organizing Unit

In visual object tracking, given the initial state of the target (its position and size in the first video frame), participants are required to analyze and process three types of data inputs: visible-light RGB image sequences, depth image sequences, and textual descriptions, in order to predict the target’s position and size in subsequent frames and achieve tracking. In the provided dataset, the RGB images and depth images are spatially and temporally aligned, while the textual data describe the target’s appearance in the first frame of the video. With the target’s position and size in the first frame given, participants are expected to design algorithms that process and analyze the tri-modal data inputs to enable efficient and accurate localization and tracking of the target.

3.2 Input and Output Description

Dataset Label Description: The target’s label in each frame is formatted as [x, y, w, h], representing the rectangular bounding box of the target in the image, where x and y are the coordinates of the top-left corner, and w and h are the width and height of the bounding box.

Algorithm Input: RGB image sequences, depth image sequences, textual descriptions, and the target’s bounding box in the first frame [x1, y1, w1, h1]. The RGB images are in three-channel UINT8 format, the depth images are in

single-channel UINT16 format, and the text TXT file contains one English sentence describing the target.

Algorithm Output: The position and size of the target in each frame of the sequence, i.e., the target's bounding box in each frame: [[x1, y1, w1, h1], [x2, y2, w2, h2], …, [xn, yn, wn, hn]].

4. Dataset and Data Description

4.1 Data Source

The dataset provided in this competition was self-collected. The visible-light RGB image sequences and depth image sequences were captured using a ZED stereo camera, and the textual descriptions were manually annotated based on the target's appearance in the first frame of the RGB image sequence. All subjects involved in the data collection process provided their consent. The data has not undergone normalization or other preprocessing operations.

4.2 Dataset Scale

The training set provided in this competition contains 1,000 sequences, each originally 600 frames long. One frame is annotated every 10 frames, resulting in a total of 60,000 RGB-Depth image pairs with target bounding boxes and 1,000 textual descriptions of the target's initial state. The validation set contains 50 sequences, comprising 11,800 RGB-Depth image pairs with target bounding boxes and 50 textual descriptions. The test sets for the preliminary, semi-final, and final rounds each contain 50 RGB-Depth image sequences and 50 textual descriptions. Sample data can be accessed via the link https://pan.baidu.com/s/1rsSJJ-PWXS1_g6m8ym_FaQ?pwd=4k8a, and the official dataset will be available for download after registration.

4.3 Data Format

The RGB image is a three-channel UINT8 image in JPEG format. The Depth image is a single-channel UINT16 image in PNG format. Text data is provided in a TXT file containing an English sentence describing the target. Each sequence's bounding box labels are in a TXT file, where each line specifies the target's position and size in a frame as [x, y, w, h].

5. Algorithm Design Requirements

5.1 Model Type

This competition does not impose strict restrictions on the type of algorithms. Participants are encouraged to use deep learning-based visual object tracking algorithms, such as networks based on convolutional neural networks (CNNs)

or Transformer architectures, and may also combine other machine learning m ethods.

5.2 Innovation

This competition encourages participating teams to propose innovative algorithmic frameworks or make creative improvements based on existing algorithms to enhance the performance of visual object tracking algorithms using multi-source heterogeneous data in complex scenarios. For example, teams may design new methods for fusing multi-source heterogeneous information and novel visual object tracking network architectures to improve tracking performance.

6. Performance Metrics Requirements

This competition uses the area under the success rate curve (AUC) as the evaluation metric for participants' solutions. The AUC is calculated as follows:

First, compute the overlap rate between the predicted target bounding box and the ground-truth bounding box:

$$R_t = \begin{cases} \dfrac{P_t \cap G_t}{P_t \cup G_t}, \ if \ P_t \neq \emptyset \ \& \ G_t \neq \emptyset \\ 1, \ if \ P_t = \emptyset \ \& \ G_t = \emptyset \\ 0, \ otherwise \end{cases}$$

Here, P_t and G_t represent the algorithm's predicted target bounding box and the ground-truth bounding box in frame t, respectively. When the target is visible in frame t, R_t measures the Intersection over Union (IoU) between the prediction and the ground-truth bounding box. If the target is out of view or fully occluded, i.e., G_t is empty, and P_t is also empty, R_t is assigned a value of 1. In all other cases, R_t is set to 0.

Next, the success rate is calculated under different thresholds. The thresholds range from 0 to 1 with a step of 0.05, i.e., $\theta \epsilon \{0, \ 0.05, 0.1, 0.15, \ldots, \ 0.95, 1\}$. The success rate at a given threshold θ_i is computed as follows:

$$u_t(\theta_i) = \begin{cases} 1, if \ R_t > \theta_i \\ 0, otherwise \end{cases}, \quad SR(\theta_i) = \frac{1}{T}\sum_{1}^{T} u_t(\theta_i)$$

Here, R_t is calculated according to the formula described above, θ_i is the threshold, and $u_t(\theta_i)$ is an indicator of whether the prediction in frame t is successful. If R_t exceeds the threshold θ_i, $u_t(\theta_i)$ is assigned a value of 1; otherwise, it is 0. The success rate at threshold θ_i denoted as SR(θ_i), is the proportion of successful frames.

Finally, by plotting the success rate SR against the thresholds θ_i, the area under the curve (AUC) is computed as the area beneath this curve.

7. Functional Requirements

7.1 Accuracy

When implementing algorithmic models for visual object tracking using multi-source heterogeneous data, high accuracy is required to ensure that the target can be tracked reliably even in relatively complex scenarios, minimizing tracking drift. Accuracy is measured using the AUC metric.

7.2 Robustness

When handling RGB, depth, and textual data of varying quality, the algorithm should operate stably and produce reliable tracking results. Even when one or more modalities have poor data quality, the algorithm should not experience significant performance fluctuations and must maintain accurate and consistent target tracking.

8. Development Environment

8.1 Software Environment

8.1.1Programming Language

This competition does not mandate the use of a specific programming language, but Python version 3.6 or above is recommended due to its extensive support for data processing, scientific computing libraries, and deep learning frameworks. Libraries such as NumPy, Pandas, and Matplotlib can be used for data processing and visualization, OpenCV for image processing, and PyTorch or TensorFlow for deep learning.

8.1.2 Deep Learning Framework

It is recommended to use TensorFlow 2.x or PyTorch 1.x, as both frameworks are widely used in the field of deep learning. They provide efficient computational performance and rich APIs, making it easier to build, train, and deploy models.

8.1.3 Computational Resources

This competition does not impose restrictions on the computational resources used by participants. Participants may use local workstations or cloud computing platforms for development and training. Local workstations should be equipped with NVIDIA GPUs (e.g., RTX series or above) to accelerate deep learning computations. Cloud platforms such as Alibaba Cloud Tianchi, Tencent Cloud TI, and Baidu AIStudio are also available, providing various configurations of computational resources, allowing participants to choose flexibly according to their needs.

8.2 Hardware Environment

This competition does not impose mandatory requirements on hardware resources such as CPU model, memory size, or GPU type. Participating teams are free to

configure their hardware according to their own needs.

9. Evaluation Criteria

9.1 Input Data Format Requirements

Participants' algorithms should be able to correctly read the competition data, including three-channel UINT8 RGB images, single-channel UINT16 depth images, TXT-format textual data, and TXT files containing the target bounding boxes [x, y, w, h].

9.2 Output Data Format Requirements

The algorithm should perform object tracking for each sequence and output the target's position and size in each frame as a TXT file. Specifically, the output should be the target's bounding boxes in each frame: [[x1, y1, w1, h1], [x2, y2, w2, h2], …, [xn, yn, wn, hn]].

9.3 Score Calculation

The scores will be determined by comparing the algorithm's output with the ground-truth annotations, using the AUC performance metric for evaluation.

10. Solution Approach

10.1 Data Preprocessing

During the training process, various data augmentation preprocessing operations can be applied. For example, RGB and depth images can be normalized, and operations such as rotation, scaling, flipping, and cropping can be performed to expand the training dataset and enhance the model's generalization capability.

10.2 Feature Extraction

Participating teams should select appropriate deep learning network architectures to extract features from RGB images, depth images, and textual data. By training these networks and combining them with machine learning methods such as feature selection, teams can achieve highly discriminative and robust data feature representations.

10.3 Model Training

Participating teams can choose suitable deep learning frameworks (e.g., TensorFlow, PyTorch) to build their models and set appropriate training parameters, such as learning rate, number of iterations, and batch size. During training, the model should be evaluated and fine-tuned using the validation dataset to prevent overfitting.

10.4 Multi-Model Decision Fusion

Participating teams may try to fuse multiple models with different architectures or training stages, using methods such as voting or weighted averaging, to combine the

predictions from multiple models and improve the final tracking accuracy. Based on the performance evaluation metrics, teams can perform targeted optimization of the models, such as adjusting the model architecture or increasing the amount of training data.

11. Reference Resources

11.1 Books

1.Deep Learning, written by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, systematically introduces the fundamental concepts, model architectures, and training methods of deep learning, and is highly helpful for understanding and applying neural networks.

2.Deep Learning with Python, authored by François Chollet, explores practical deep learning using Python, covering applications such as computer vision, natural language processing, and generative models. The book includes more than 30 code examples with detailed step-by-step explanations.

11.2 Online Courses

1.The Deep Learning Specialization course on Coursera, taught by Professor Andrew Ng, covers several key areas of deep learning, including neural network fundamentals, convolutional neural networks, and recurrent neural networks. The course is rich in content and highly theoretical.

2.The Hands-on Deep Learning with PyTorch course on Bilibili explains how to implement deep neural networks using Python and the PyTorch deep learning framework, with strong emphasis on practical applications.

11.3 Academic Papers

[1] Zhou L, Zhou Z, Mao K, et al. Joint visual grounding and tracking with natural language specification[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 23151-23160.

[2] Zhu J, Lai S, Chen X, et al. Visual prompt multi-modal tracking[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 9516-9526.

[3] Zhu X F, Xu T, Liu Z, et al. UniMod1K: towards a more universal large-scale dataset and benchmark for multi-modal learning[J]. International Journal of Computer Vision, 2024, 132（8）: 2845-2860.

12. Submission Requirements

12.1 Algorithm Code

Participating teams must submit the complete algorithm code, including all

components such as data preprocessing, model training, and prediction/inference. The code should be written in standard-compliant **Python**, with clear comments and documentation to ensure that the reviewers can understand and execute it.

12.2 Test Result Files, Model Files, and Environment

Participating teams must submit the test result files and the trained model files (since models may be large, please provide a download link). Additionally, teams should provide instructions for loading and using the model, including the required runtime environment and dependency libraries. The model files must be able to run properly in the designated test environment and output prediction results.

12.3 Technical Report

Participating teams must submit a detailed technical report, which should include the algorithm design approach, model architecture diagrams, experimental setup (such as training parameters and data augmentation methods), performance analysis (a detailed analysis of primary and secondary metrics), as well as an analysis of the innovations and limitations of the algorithm. The technical report should be in PDF format, with no restriction on the number of pages.

13. Updates and Q&A

The competition may be updated over time. To address questions that participants encounter during the contest, a dedicated Q&A group will be set up for participants (visible only after official registration).

14. Competition Process and Award Settings

14.1 Registration Stage

Participants complete registration on the official competition website, submit individual or team information, and obtain the download link for the preliminary stage dataset.

14.2 Preliminary Stage

Participants use the training dataset provided by the competition organizers to design their algorithm models and validate and debug their methods using the preliminary stage test set. During the preliminary stage, there is no limit on the number of daily submissions, but the preliminary leaderboard is updated every hour.

14.3 Registration Stage

After the preliminary stage, the competition enters the semifinal stage, and the download link for the semifinal dataset is made available. Only teams that submitted valid results in the preliminary stage are eligible to enter the semifinals. During the

semifinal stage, participants use the provided semifinal dataset to debug their algorithm models and submit test results for the semifinal test data. The semifinal stage lasts for three days, and each team can submit only twice per day. The semifinal leaderboard is updated every hour.

14.4 Semifinal (Provincial Competition) Results Announcement

The semifinal results are announced on the official competition website. Using the number of teams that entered the semifinals as the basis for awarding, first, second, and third prizes for the semifinals are selected according to the provincial competition award ratio, with corresponding provincial competition certificates issued. During the award selection process, any submission with algorithm performance below the baseline reference score provided by the organizers will be considered invalid and not eligible for awards. Teams winning the first and second prizes in the semifinals will advance to the national final.

14.5 Final Stage

1.Online Final Evaluation: During the final stage, the download link for the final dataset is made available. Participants use the provided final dataset to debug their algorithm models and submit test results for the final test data. The final stage lasts for one week, and each team can submit only once per day. The final leaderboard is updated every hour. Based on the final leaderboard results, using the number of teams entering the finals as the basis for awarding, a list of national first prize candidates and the winners of the national second and third prizes is selected according to the national competition award ratio, with certificates issued for the second and third prizes.

2.Final Submission: Teams nominated as national first prize candidates must submit, within the specified time, technical documentation, algorithm code and model files, demonstration videos, and supplementary materials. No modifications or additional submissions will be accepted after the deadline.

3.Final Review Stage: A professional review panel will reproduce and verify the submissions of the national first prize candidate teams. If there are any questions during the review, participants may be asked to provide explanations.

4.Offline National Final: National first prize candidate teams must submit their finalized technical documentation, algorithm code and model files, demonstration videos, and supplementary materials within the specified time, and participate in the offline national final review and defense. The national first prize winners and their rankings are determined based on algorithm performance scores and offline defense

scores. Teams that do not participate in the offline review and defense will be considered as forfeiting the award. Certificates of honor will be awarded to the national first prize winners.

15. Contact Information

Competition Communication QQ Group: 1011080681

Email: xuefeng.zhu@jiangnan.edu.cn

Registration Website: www.aicomp.cn