Global Instance Tracking: Locating Target More Like Humans

Shiyu Hu, Xin Zhao, *Member, IEEE*, Lianghua Huang, and Kaiqi Huang, *Senior Member, IEEE*

Abstract—Target tracking, the essential ability of the human visual system, has been simulated by computer vision tasks. However, existing trackers perform well in austere experimental environments but fail in challenges like occlusion and fast motion. The massive gap indicates that researches only measure tracking performance rather than intelligence. How to scientifically judge the intelligence level of trackers? Distinct from decision-making problems, lacking three requirements (a challenging task, a fair environment, and a scientific evaluation procedure) makes it strenuous to answer the question. In this article, we first propose the *global instance tracking (GIT)* task, which is supposed to search an arbitrary user-specified instance in a video without any assumptions about camera or motion consistency, to model the human visual tracking ability. Whereafter, we construct a high-quality and large-scale benchmark *VideoCube* to create a challenging environment. Finally, we design a scientific evaluation procedure using human capabilities as the baseline to judge tracking intelligence. Additionally, we provide an online platform with toolkit and an updated leaderboard. Although the experimental results indicate a definite gap between trackers and humans, we expect to take a step forward to generate authentic human-like trackers. The database, toolkit, evaluation server, and baseline results are available at http://videocube.aitestunion.com.

Index Terms—Global instance tracking, single object tracking, benchmark dataset, performance evaluation, human tracking ability

1 Introduction

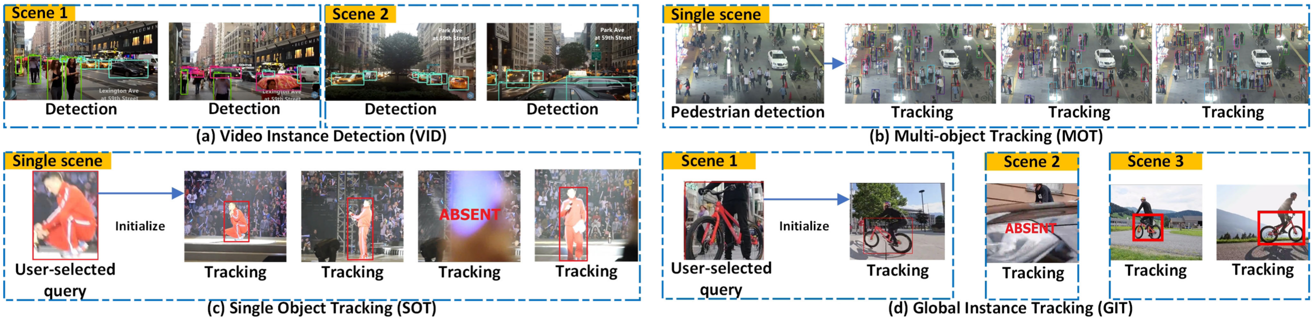
Target tracking, the ability to follow a moving object with the human eyes, is the basic function of the human visual system. Research reveals that a baby can master this skill at only a few weeks of age and quickly expand from tracking salient objects (e.g., a brightly colored toy) to arbitrary objects (e.g., a decoration on the clothes of parents) [6], [7]. Inspired by the powerful human visual system and eye-catching artificial intelligence technology, researchers have proposed a series of visual tasks to locate moving targets in the real environment. Several existing computer vision tasks, such as single object tracking (SOT [8]), multi-object tracking (MOT [9]), and visual instance detection (VID [10]), simulates human target tracking ability to locate moving targets in the natural environment, and are widely used in animal behavior observation [11], [12], [13], [14], medical research [15], [16], [17] and robot navigation [18].

However, challenging conditions like occlusion, fast motion, and weak illumination reduces the performance of existing methods. Take automatic driving as an example - several crashes happened at night or under bright light conditions due to the limit in visual perception robustness of trackers, which contrasts to the high performance judged by the vision task benchmarks. In other words, existing experimental environments only measure performance rather than intelligence, far away from the actual applications. A natural question is, how to scientifically measure the tracking intelligence of an algorithm?

The imitation game proposed by Alan Turing in 1950 [19], which is usually called the Turing test, is a recognized standard to judge machine intelligence. Recently, the agents represented by AlphaGo (Go game [20] AI) and DeepStack (Poker game [21] AI) have defeated the top human professional players in decision-making problems, and become the landmark results of the Turing Test. From these works, we can summarize three requirements for machine intelligence measurement: (1) a challenging task (e.g., Go game is difficult for both humans and machines); (2) a fair competition environment (e.g., human and machine compete in the Go game with equal rules); and (3) a scientific evaluation procedure (e.g., players with a larger number of vacant

- *Shiyu Hu is with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Center for Research on Intelligent System and Engineering, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: hushiyu2019@ia.ac.cn.*
- *Xin Zhao is with the Center for Research on Intelligent System and Engineering, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: xzhao@nlpr.ia.ac.cn.*
- *Lianghua Huang is with the Center for Research on Intelligent System and Engineering, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: huanglianghua2017@ia.ac.cn.*
- *Kaiqi Huang is with the Center for Research on Intelligent System and Engineering and National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with the CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing 100190, China. E-mail: kqhuang@nlpr.ia.ac.cn.*

Manuscript received 30 May 2021; revised 20 January 2022; accepted 15 February 2022. Date of publication 23 February 2022; date of current version 5 December 2022.
This work was supported in part by the National Natural Science Foundation of China under Grants 61721004 and 61876181, in part by the Projects of Chinese Academy of Science under Grant QYZDB-SSW-JSC006, in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27000000, and in part by the Youth Innovation Promotion Association CAS.
(Corresponding author: Xin Zhao.)
Recommended for acceptance by H. Kjellstrom.
Digital Object Identifier no. 10.1109/TPAMI.2022.3153312



(a) Video Instance Detection (VID)

(b) Multi-object Tracking (MOT)

(c) Single Object Tracking (SOT)

(d) Global Instance Tracking (GIT)

Task	Basic task	Object		Scenario		Arbitrary instance	Random scenario
		Object category	Obejct selection	Motion consistency assumption	Scene-changing		
VID	Detection	Limited by training set	Detected	N	Y	N	Y
MOT	Detection & Tracking	Limited, usually pedestrians or vehicles	Detected in the first frame	N	N	N	N
SOT	Tracking	Arbitrary	User-specified in the first frame	Y	N	Y	N
GIT	Tracking	Arbitrary	User-specified in the first frame	N	Y	Y	Y

(e) Comparison of GIT task with other video-related tasks

Fig. 1. The execution flow and comparison of GIT with other video-related vision tasks (VID, MOT, SOT). VID (*a*) and MOT (*b*) can only locate limited instances, while SOT (*c*) and GIT (*d*) do not constrain the target category. Furthermore, GIT expands the SOT task by canceling the motion continuity assumption, allowing the target to move in a broader and more complex environment. The detailed comparison of GIT with above vision tasks is listed in *e*. Obviously, GIT is a new visual task without restrictions on target categories and scenarios.

intersections and captured stones win the Go game). Nevertheless, the existing target tracking area lacks these three points, making it strenuous to evaluate visual intelligence.

For the first requirement, a proper task is essential to estimate visual tracking intelligence. Simple assignments (such as tracking a black dot on a white screen) cannot reflect intelligence, while unmanageable tasks (such as tracking an ant in a colony with a shaking camera) are almost impossible for humans to execute. Therefore, the reasonable idea is to design a moderately difficult task based on human visual tracking ability. Clearly, people can unconsciously locate an *arbitrary instance* in *random scenarios*, while the existing tasks always contain strong constraints on target categories (MOT, VID) or scenarios (SOT).

As the second requirement, a suitable benchmark needs to reflect the characteristics of the task and simulate the natural environment. Dynamic visual acuity, the essential human ability to perceive moving objects, can be improved by tracking fast-moving targets in complex environments. Thus, a decent benchmark should fitly reproduce the proximate real-world conditions and provide a platform for training a human-like tracker. However, existing tracking benchmarks only provide a simplistic environment. Trackers generated by these benchmarks are still far from the human visual system and cannot suit challenging realistic conditions like occlusion, fast motion, and weak illumination.

The last requirement, a scientific evaluation system, should set targets (machine and human) into the same environment and measure their tracking capabilities with reasonable indicators. Unlike Go and poker games with clear rules, trackers and humans have exceptionally distinct ways of performing visual tracking tasks. Algorithms usually process the video frame by frame and return bounding boxes to locate the object, while humans directly focus their sight on the target. Existing benchmarks are all designed for evaluating algorithms but lack standards for measuring human tracking ability. Lacking the comparison with humans means we cannot measure the intelligence level of algorithms accurately.

Based on the above three problems, this work evaluates tracking intelligence degree for the first time by providing:

(1) A proper task to model human visual tracking ability. We introduce *global instance tracking (GIT)*, a new task of searching an arbitrary user-specified instance in a video without any assumptions on camera or motion consistency, to accurately model the human tracking ability. Unlike the existing video-related tasks, GIT aims to find all video fragments where a query object presents and locates its trajectories in these fragments. GIT retains the category-independent advantage and expands the boundary of the traditional SOT task to approach object tracking in general scenes. An ideal GIT algorithm is supposed to work in different video environments like rapid view angle changes, frequent camera switches, or long-term target absences. The execution flow and comparison of GIT with other video-related vision tasks are shown in Fig. 1.

(2) A comprehensive benchmark to simulate the real world.We provide a high-quality, large-scale benchmark *VideoCube* for this novel task. It consists of 500 long-term videos that cover different object classes, scenario types, motion modes, and challenge attributes, with an average length of *14920* frames. Figs. 2 to 4 illustrates that by comparing with existing visual tracking benchmarks, VideoCube provides a proximate real-world environment and evaluates the algorithms scientifically.

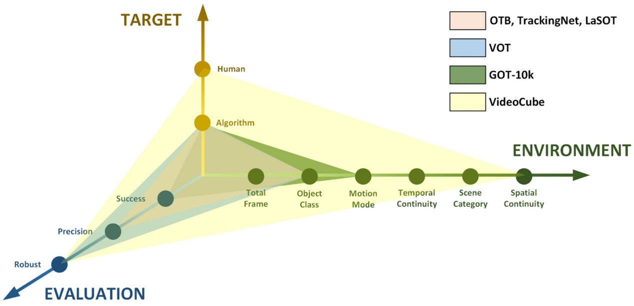
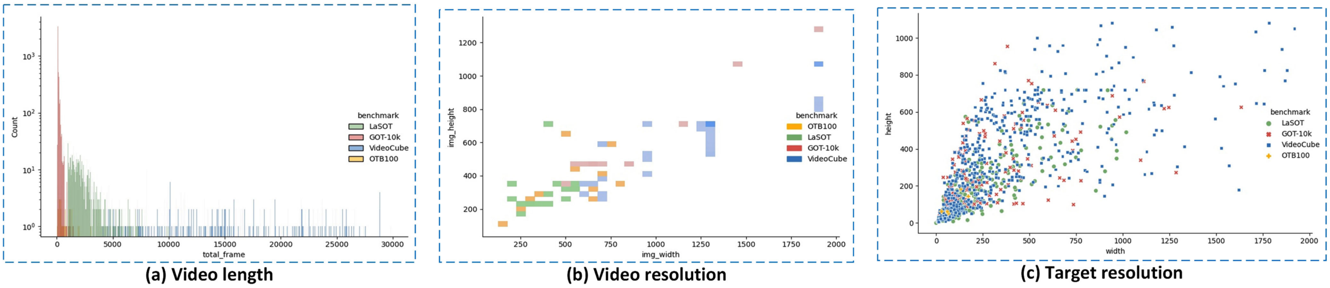


Fig. 2. Comparison of VideoCube and other tracking benchmarks (OTB2015 [1], TrackingNet [2], LaSOT [3], VOT2017 [4], GOT-10k [5]) in the complexity of the environment, the rationality of evaluation, and the completeness of target selection.

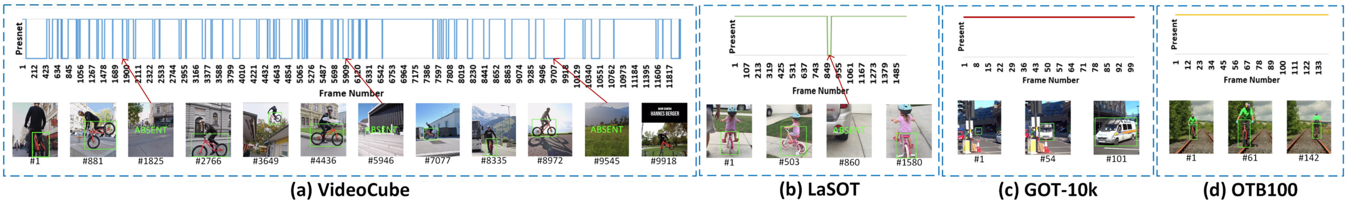


(a) Video length

(b) Video resolution

(c) Target resolution

Fig. 3. Comparison of VideoCube and three representative tracking benchmarks (OTB2015 [1], LaSOT [3], GOT-10k [5]) in video length (*a*), video resolution (*b*), and target resolution (*c*).



(a) VideoCube

(b) LaSOT

(c) GOT-10k

(d) OTB100

Fig. 4. Comparison of VideoCube (*a*) and three representative tracking benchmarks (LaSOT [3] (*b*), GOT-10k [5] (*c*), OTB2015 [1] (*d*)) in video content, video length, number of disappearances, and the absent duration.

(3) A scientific evaluation procedure to compare humans and machines with reasonable indicators. In addition to evaluating trackers via classical metrics, we judge *human visual tracking capability* via an eye-tracking experiment for the first time. Fig. 5 is the schematic diagram of the human visual tracking experiment. Human performance is treated as a baseline to measure the intelligence level of existing methods. The result illustrates that SOTA trackers can perform well in a simple situation (target with smooth movement) but fail in difficulties (e.g., occlusion, fast motion, and weak illumination), while humans can still maintain fast and accurate tracking with challenging factors.

Besides, we provide a comprehensive online platform at http://videocube.aitestunion.com with systematic evaluation toolkits, an online evaluation server, and a real-time leaderboard. We believe the online platform with the human baseline can provide researchers with more comprehensive assistance in visual intelligence research and take a step forward to generate authentic human-like trackers.

The rest of this paper is organized as follows. Section 2 provides a review of video-related tasks and distinguishes them from GIT. Section 3 introduces the design principles of VideoCube. The experimental results and detailed analysis are described in Section 4. Finally, we conclude this paper and discuss future works in Section 5.

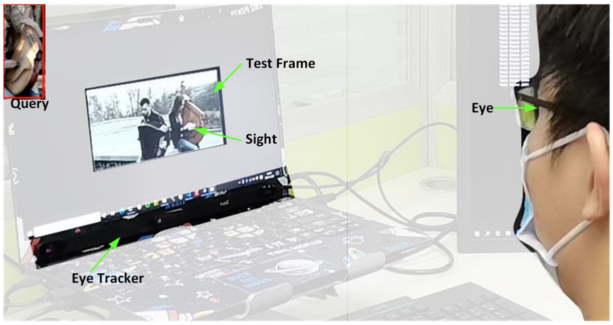


Fig. 5. Schematic diagram of the human visual tracking experiment.

2 Related Work

Capturing local motion and predicting long-term moving trajectories of targets in a video is of great significance to many research fields [15], [22]. Several vision tasks have been modeled for locating moving objects in video. This section introduces these visual tasks' definitions, characteristics, and application scenarios to distinguish them from GIT.

2.1 Locate Specific Target Categories in Random Scenarios

Video instance detection (VID) [10] is a fundamental prerequisite for advanced visual tasks such as scene content analysis and understanding. It aims to accurately determine the category and location of each target in a video. The target category is generally limited to the known classes in the training dataset, but the video without any restrictions may contain various scenes.

Multiple object tracking (MOT) [9] is a model-specific visual task that focuses on tracking specific categories like persons or vehicles without any prior knowledge about the appearance and amount. The general MOT algorithm usually runs a detector to obtain the object's bounding box in the first frame and generate features; then calculates the similarity to determine instances belonging to the same target and assigns a digital ID to each object.

2.2 Track Random Objects in a Single Scenario

Single object tracking (SOT) [8] intends to calculate the location of a user-specific visual target in the video when only a position in the first frame is available. Unlike other visual tasks, SOT is an entirely category-independent assignment suitable for open-set testing with broad prospects. However, the implicit motion continuity assumption limits its actual applications. Since SOT is the vision task closest to GIT in assignment settings, the following part introduces the related trackers and benchmarks in detail.

2.2.1 Trackers

Correlation-Filter Trackers. Correlation-filter (CF) trackers regard the SOT task as a regression problem and achieve high speed via fast Fourier transform (FFT) [23]. Dense image sampling by circulant shift on a single centered image patch is essential to implement fast training and inference in the Fourier domain. As the first model to utilize the correlation filter framework in object tracking, MOSSE [24] considers this task as a regularized least-squares problem and reformulates its closed-form solution, achieving reliable tracking performance at 700 fps. Later on, several improvements have been proposed, including using a scale embedding to handle scale variation [25] and improving CF tracking via extra regularization method [26].

Deep Trackers. Recently, several methods based on deep learning have been proposed to advance tracking performance. Convolutional neural networks (CNNs) are the most widely-used model, involving extracting features through pre-trained models [27] or using end-to-end learning to generate object appearance models [28]. The siamese trackers [28], [29] and their variants [30], [31] regard tracking as a feature matching task and achieve a significant result. By learning a high dimensional metric space between the exemplar and search patches, siamese trackers can quickly localize the instance in a consecutive sequence. Except for CNN-based models, some advanced deep trackers regard tracking as a sequential decision-making task [32], or combine the recurrent structures to accomplish sequential prediction [33].

2.2.2 Benchmarks

Short-Term Tracking Benchmarks. A series of benchmarks have appeared since 2013 and provide a consolidated platform for evaluating and analyzing algorithms. As one of the earliest benchmarks, OTB2013 [34] includes 51 fully-labeled short sequences and evaluates the performance of the previous 29 top trackers. Subsequently, OTB2015 [1] expands the benchmark to 100 videos to provide unbiased performance comparisons. The VOT [4], [35], [36], [37], [38], [39], [40] has been an annual visual object tracking challenge since 2013, which provides a diverse and adequately small dataset from existing visual tracking datasets. TC-128 [41] collects and annotates 78 new videos based on OTB2013 [34] to provide the evaluation of color-enhanced tracking algorithms on color sequences. Several datasets are designed for tracking specific instances. The NUS-PRO [42] dataset focuses on tracking pedestrian and rigid objects, and the UAV123 [43] comprises 123 short videos for assessing unmanned aerial vehicle tracking performance. Nfs [44] provides 100 sequences with a higher frame rate (240 FPS) camera, intending to examine the trade-off bandwidth limitations related to real-time analysis of visual trackers. With the advancement of deep learning, a large-scale and high-quality dataset for short-term tracking is demanded. GOT-10k [5] is a significant high-diversity benchmark and comprises 10,000 videos from the semantic hierarchy of WordNet [45] to accommodate plentiful object categories and motion trajectories. It is the first benchmark to suggest the one-shot protocol for evaluating tracking performance and improving model generalization.

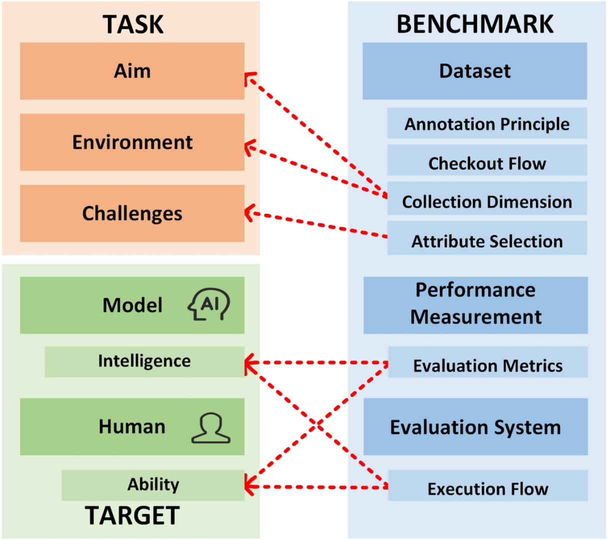


Fig. 6. Construction principles of the VideoCube benchmark. We assume that a scientific benchmark should characterize the specified task and evaluate the model intelligence. Dataset, evaluation system, and performance measurement are three critical points included in constructing a benchmark. The red dotted line expresses the relationship of various fields.

Long-Term Tracking Benchmarks. Allowing brief disappearance and having a longer duration are two characteristics of long-term tracking. OxUvA [46] is the first large-scale dataset for this task and selects 366 videos with an average duration of 144 seconds, but only performs annotation every 30 frames. LaSOT [3] is first released in 2019 and provides a dataset with 3.5M manually labeled frames, including 1400 videos with 70 categories. In 2020, LaSOT is expanded to 1550 videos and 85 classes. It is re-divided with the one-shot protocol of GOT-10k [5] to improve the generalization.

Consequently, SOT can continuously locate objects of any category due to model-free characteristics and is more versatile for open-set test environments. However, the existing SOT methods are still far from robust long-term tracking in complex environments for three reasons:(1) Strong constraints in the task definition. The implicit continuous motion assumption limits the task environment in continuous-time and single-scene, far from the natural application environment. (2) Limited video type in the existing benchmarks. Videos with a single shot and a single scene cannot fully reflect the complexity of the actual situations. (3) Strong timing-dependence in the modeling process, which accumulates errors and cannot achieve robust tracking in long-term tracking.

3 Construction of VideoCube

As a high-quality benchmark, VideoCube contains a large-scale dataset, reasonable evaluation metrics, and scientific evaluation systems to provide a general platform for intelligence measurement (Fig. 6).

3.1 Dataset

VideoCube is a reliable global instance tracking benchmark that contains scenes and instances adequately to reflect the

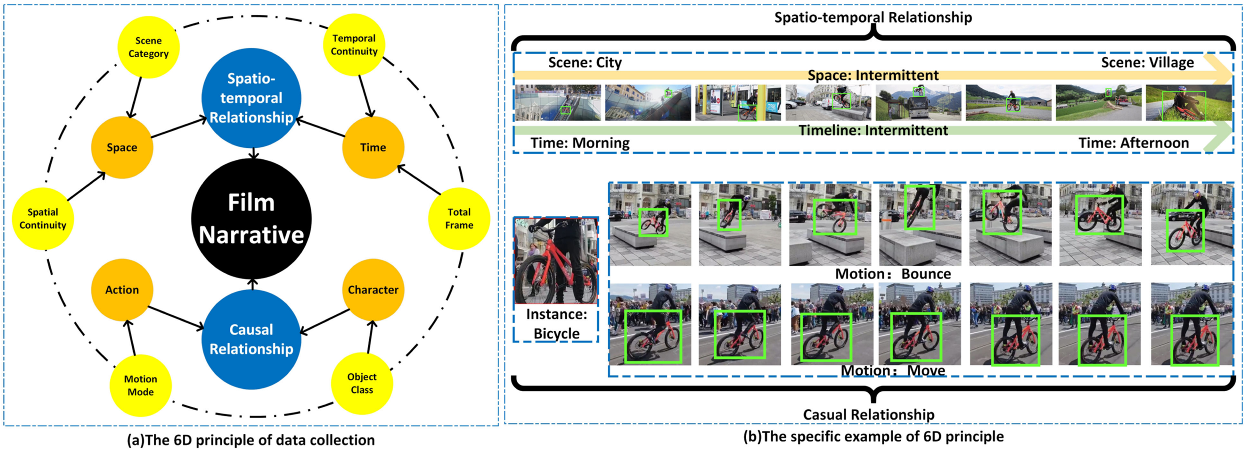


Fig. 7. The 6D principle of data collection. We split the film's narrative into the spatio-temporal and causal relationship and further decompose them into six dimensions (scene category, spatial continuity, temporal continuity, total frame, motion mode, object class) to provide a more comprehensive description.

diversity of real life. Before constructing it, we first summarize the key elements (e.g., benchmark, task, and target) and propose our design principles based on Fig. 6. Several aspects are considered in constructing this dataset:

(1) Multiple Collection Dimension. The collection of VideoCube is based on six dimensions (Fig. 7) to describe the spatio-temporal relationship and causal relationship of film narrative, which provides an extensive dataset for the novel GIT task. We guarantee that each video contains at least *4008* frames, and the average frame length in VideoCube is around *14920*. Besides, the selected videos contain *transitions* and target *disappearance-reappearance* process to cancel the motion continuity assumption.

(2) Specific Annotation Principle and Exhaustive Checkout Flow. A professional labeling team manually marked each video with a 10Hz annotation frequency, and all videos have passed three rounds of review by trained verifiers. Based on rigorous experiments, we selected the most effective algorithm PrDiMP [48] to combine manual annotations and accomplish intensive labels with 30Hz frequency.

(3) Comprehensive Attribute Selection. Multiple shots and frequent scene-switching make the video content change dramatically and become more challenging for algorithms. Thus, we accommodate twelve attributes annotations for each frame to implement a more elaborate reference for the performance analysis.

3.1.1 Collection Dimension

The collection dimension is an essential basis for constructing datasets. Rich dimensions can restore the narrative content and simulate real application scenarios through dimensions integration. However, most existing video datasets only consider instance category and video duration when constructing but lack an overall narration expression. As the scale of datasets has increased in recent years, several datasets have begun to extend their collection dimensions. For example, GOT-10k [5] combines the motion modes, and LaSOT [3] adds a natural language description to characterize the video content. Nevertheless, we consider that the existing datasets lack a widespread meditation on dimension selection. The organization of instance categories and motion modes such as GOT-10k [5] is suitable for short-term rather than long-term tasks. The natural language description used by LaSOT [3] seems to express the video content intuitively, but this annotation is subjective since personal views will inevitably be involved. Besides, an extra algorithm is needed to extract useful information in sentences, which increases the complexity of usage and errors.

How to determine the collection dimensions? The film narrative is defined as a chain of causal relationship events occurring in space and time [51]. The causal relationship is determined by characters and activities, while the spatio-temporal relationship combines scene, time, and their continuity. Consequently, we connect scene category, spatial continuity, temporal continuity, total frame, motion mode, and object class as *6D principle* (Fig. 7) to collect videos in VideoCube. The detailed introduction of 6D principle is organized as follows:

Object Classes. Different from the existing datasets, VideoCube collects 89 typical instances and divides them into nine main categories based on the semantic framework WordNet [45]. As shown in Fig. 9a, it maintains an even distribution across the main categories. Since *person* is the most common instance category while people with different identities have significant differences in motion modes and appearances, we split the person class into performer, athlete, and other careers. Besides, given that computer-generated instances are common in some application scenarios but ignored by other datasets, we also add the functional character.

As shown in Figs. 10a and 10b, VideoCube has advantages in the distribution of object classes, and the nine root categories maintain uniform distribution. Although LaSOT [3] maintains an even distribution on 70 classes, half of the data belong to the *animal* category, while only 20 sequences (1.43%) belong to the *person*.

Spatial Continuity and Scene Categories. Videos in VideoCube are divided into normal space and fictional space. First of all, 56 videos (to keep the same video amount with the fictional character in Fig. 9a) are reserved as the fictional space. Since VideoCube cancels the motion continuity assumption, the instance may occur in multiple scenes, causing scene-switching in a video. Therefore, we divide the normal space

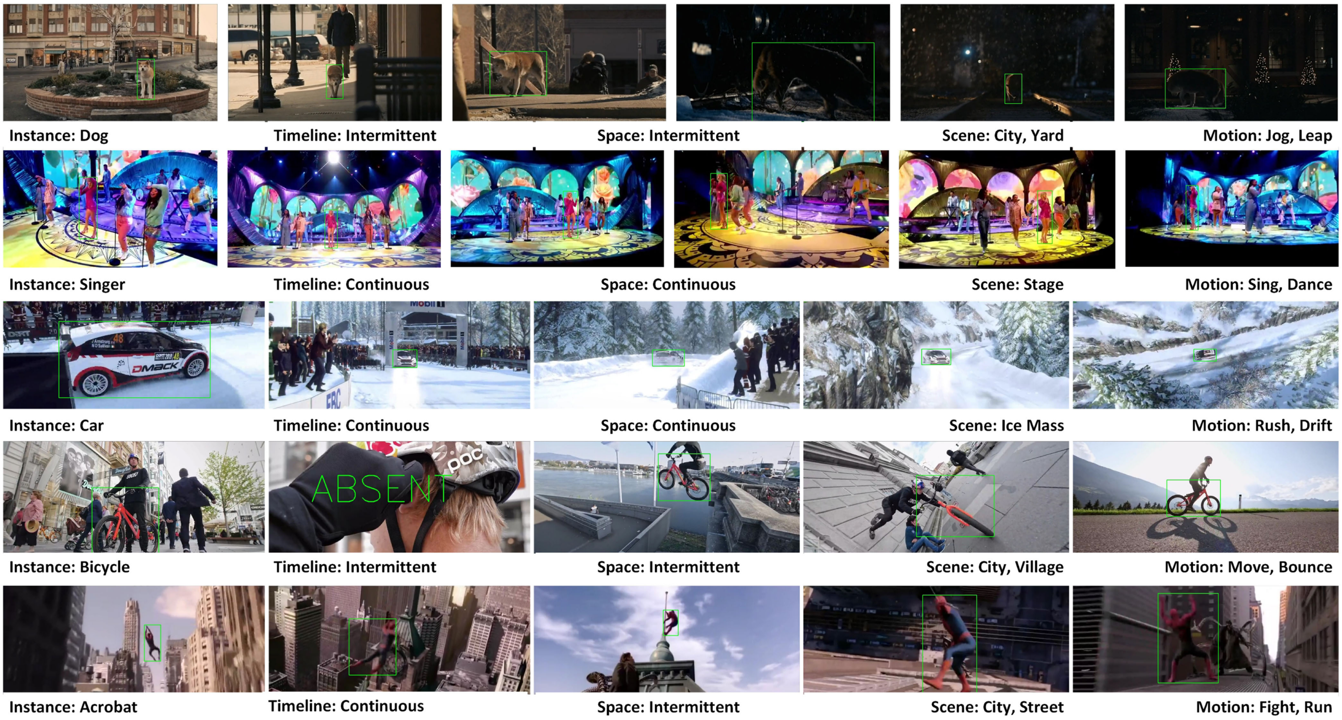


Fig. 8. The representative data of VideoCube. Each video is strictly selected based on duration, instance classes, main scene categories, main motion modes, spatial consistency, and time consistency.

videos into 222 continuous spaces and 222 intermittent spaces, then record the single scene of continuous space and two main scenes of intermittent space. Finally, all the 666 scenes are evenly divided into seven main categories, as shown in Fig. 9b.

Figs. 10c and 10d exhibits the distribution of scene categories in VideoCube and LaSOT [3]. Since the object class of LaSOT [3] is mainly animals, its scene categories are primarily concentrated in outdoor scenes.

Temporal Continuity and Video Duration. From a temporal perspective, VideoCube divides 500 videos into time-continuous and time-intermittent. Canceling the continuous motion hypothesis breaks the temporal boundaries and extends the proportional timeline to a flexible one. For example, a 3-minute video of the SOT task can only record a 3-minute event. In contrast, a 3-minute video can be edited to reflect a story for more than an hour in the GIT task, increasing the richness of video content. As shown in Fig. 9c, video duration in VideoCube can be equally divided into four categories ranging from 3 minutes to 20 minutes, which is much higher than the existing video-based datasets.

Motion Modes. VideoCube records the two principal motion modes for each video. The 1000 motion modes are divided into 61 categories, as shown in Fig. 9d.

Figs. 10e and 10f shows the distribution of motion modes in VideoCube and LaSOT [3]. Obviously, the total number of motion modes in VideoCube (61) is much larger than LaSOT (33). Besides, the statistical results of LaSOT are mainly concentrated in the most common modes, while the statistical results of VideoCube are distributed in a variety range. The long tail of distribution results in VideoCube indicates our work includes more rare movements.

We believe that the 6D principle provides a scientific guide for the data collection, which increases the richness of video content and helps users quickly restore the narration from the six elements, improving the practicality of VideoCube. Fig. 8 illustrates the representative frames of this dataset.

3.1.2 Annotation Principle

We use manual labeling and automatic algorithm for data annotation. The professional annotation team manually labels every three frames at a frequency of 10 Hz. After that, the PrDiMP [48] algorithm automatically provides labels for the rest two frames between manually labeled frames, as shown in Fig. 13.

Manual Annotation. A professional project team rigorously labels VideoCube. The annotation process observes the following rules: (1) if the specific instance appears in the frame, the visible part of the instance is marked with the tightest bounding box; (2) if the instance is not in the frame, an absent label is marked. Besides the two main rules above, some exceptions require individual labeling rules. We summarize the exceptional cases of high-frequency occurrences. The examples are provided in Fig. 11a: (1) *Tiny area*: if the instance is divided into multiple areas by obstacles and labeling the tiny area will contain many obstacle pixels, the tiny part is discarded, and only labels the central area (Figs. 11a, 11b and 11c). (2) *Transparent objects*: transparent beards of cats or mice are not marked (Figs. 11b and 11d). (3) *Slender and broad swinging objects*: the tail of a mouse or a long ribbon of a person are not marked (Figs. 11d and 11e).

VideoCube also provides the instance absent label, the occlusion label, and the starting points of all shots. The transition is divided into two types: *fast transition* and *slow transition*. Transitions that occurred in two successive frames without motion continuity are considered as *fast transition*,

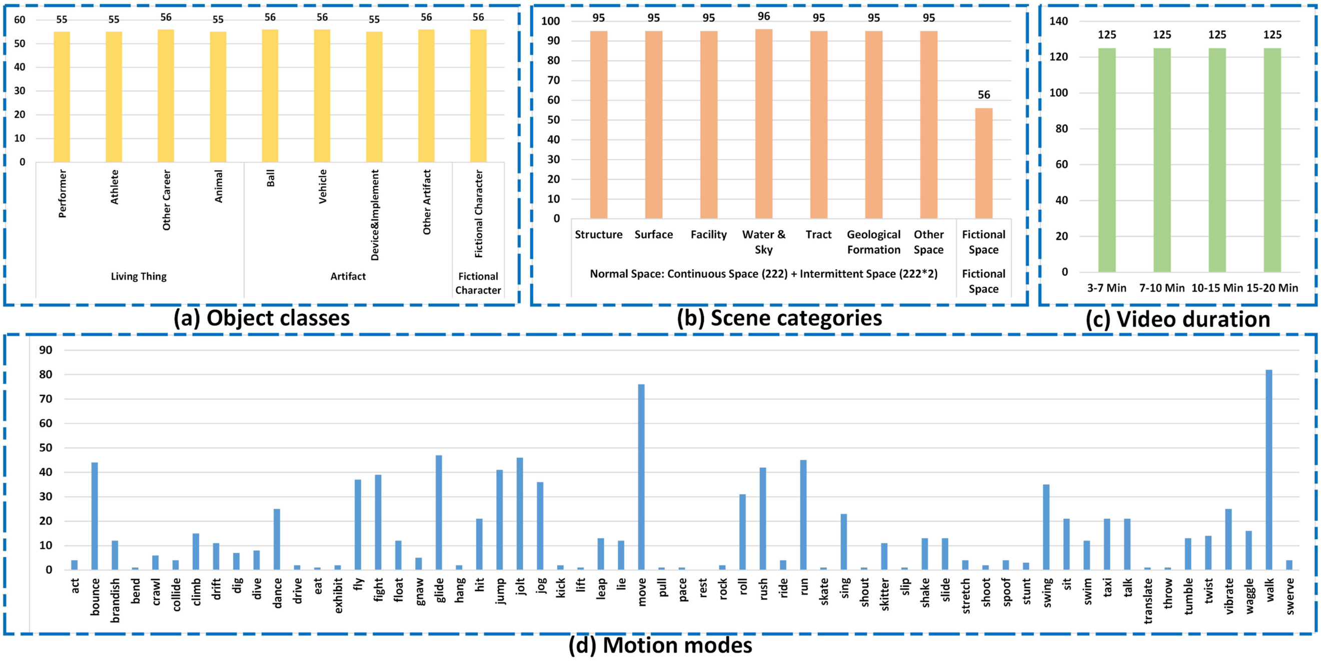


Fig. 9. Data distribution of VideoCube. (*a*) The distribution of object classes. (*b*) The distribution of scene categories. (*c*) The distribution of video duration. (*d*) The distribution of motion modes.

and the start frame of each new shot is labeled as a *shot-cut*. Dissolve, fade-in, and fade-out between two scenes are *slow transition*, and all frames belonging to the interim stage are labeled. Examples of transitions are shown in Fig. 12.

Automatic Annotation. The execution steps and completion strategies of the automatic annotation algorithm are shown in Fig. 13. The red dashed box represents a manually annotated frame, while the green dashed box represents an automatically completed frame. For the first row, the annotation team labels #606 and #609, and records the target position. Since the shot-switching occurs from #608 to #609, an extra transition tag is needed for #609 to indicate the beginning of a new shot. The second row explicates the process of labeling #643 via PrDiMP [48]. The target position of two nearest frames with manual annotation is marked as *gt-past* (#642) and *gt-next* (#645). In this sequence, PrDiMP [48] runs twice with forward order (from #642 to #645) and backward order (from #645 to #642) and records target location as *positive* and *negative*. We design several strategies to synthesize the position parameters of positive, negative, gt-past, and gt-next for different situations, then obtain the coordinate of instance in #643. Algorithm 1 presents the framework for generating the automatic labels.

To verify the effect of the above strategy, we select LaSOT [3] as the experiment dataset. It is a large-scale, long-term tracking dataset with a 30 Hz manual label frequency (provide the manual label for each frame). The first version is released in 2019 with 1400 videos (total of 3.5M frames), while the new version in 2020 is expanded to 1550 videos (total of 3.87M frames). In this experiment, we select the first version to verify the performance of the automatic label method. Fig. 14 presents the experimental result of the automatic annotation on LaSOT. The blue line in Fig. 14a represents 1 Hz manual annotation frequency with the automatically generated result for the middle 29 frames; the orange line represents 10 Hz manual annotation frequency with the automatically generated result for the two middle frames. The average IoU score based on the 1 Hz complementation plan is 0.9, while the average IoU score based on the 10 Hz is 0.95. Fig. 14b shows the IoU value of all 1400 videos based on 10 Hz manual annotation frequency with the automatically generated result for the two middle frames. It indicates that the 1 Hz manual labeling frequency is too sparse to provide a factual basis of the automatic completion scheme (such as TrackingNet) or evaluation (such as OxUvA). The 10 Hz manual labeling frequency with a suitable automatic annotation mechanism can improve efficiency and provide a human-level annotation via an effective algorithm.

3.1.3 Checkout Flow

We implement a strict data review process to ensure the quality of the benchmark. The construction process is divided into two tasks: data collection and data annotation. Professional collectors and annotators are trained to comprehend the GIT task's characteristics and complete the preliminary work with a self-inspection process. The verifiers review the submitted data as the second-round verification. Finally, the authors judge whether to accept or reject it in the third-round confirmation. As shown in Fig. 15, any rejection in self-check, verification, or data acceptance will result in the re-collection. We believe the three-round verification mechanism can generate a high-quality dataset.

3.1.4 Attribute Selection

Twelve attributes are selected in VideoCube to enable further performance analysis:

Instance Absent (IA), the instance is out-of-view or fully occluded by other objects, manually labeled by annotators.

Shot-cut (SC), frame belongs to slow transition or fast transition, as the starting point for a new shot and is manually labeled by annotators.

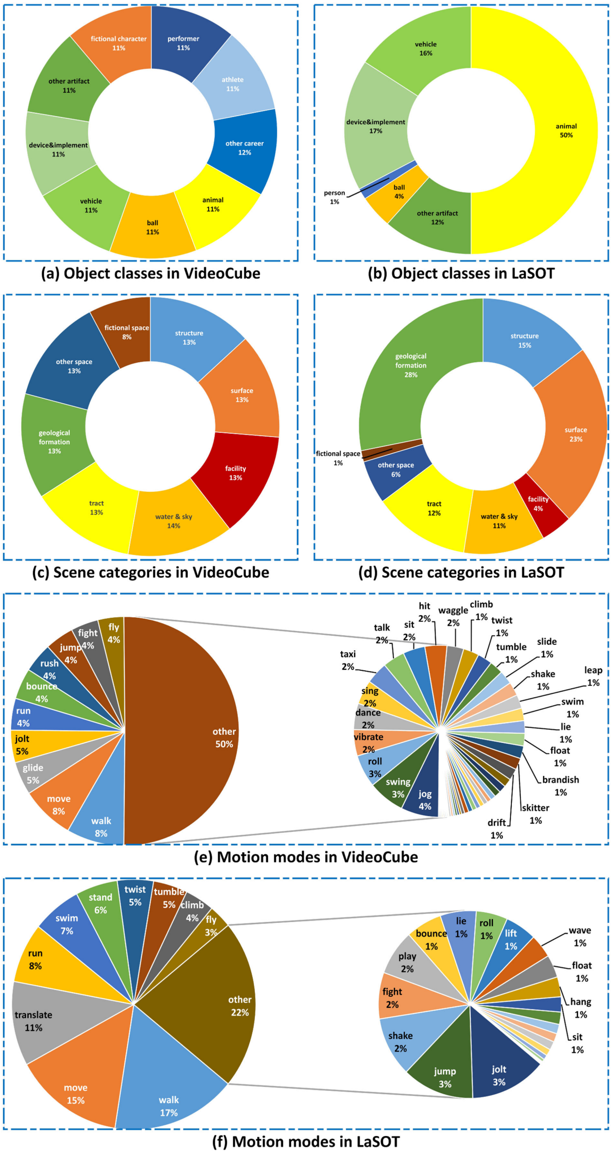


Fig. 10. The distribution of object classes (*a-b*), scene categories (*c-d*) and motion modes (*e-f*) in VideoCube and LaSOT [3] (based on WordNet [45]).

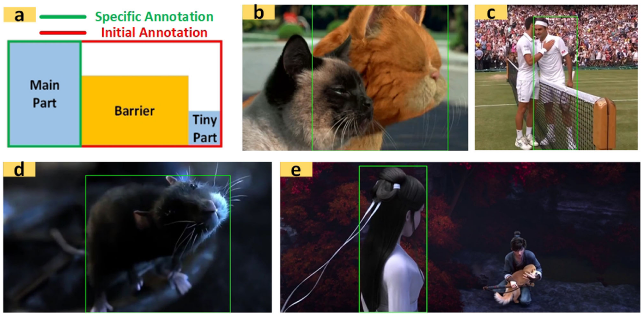


Fig. 11. Examples of specific rules in VideoCube annotations. (*a*) Example of a tiny area. (*b*)Garfield's transparent beard and a tiny part of the left side. (*c*)Federer's right hand. (*d*)The mouse's transparent beard and a slender tail. (*e*)The white long ribbon.

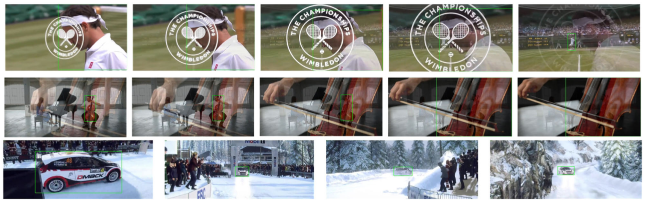


Fig. 12. Examples of transitions in VideoCube annotations. (The first and second rows belong to the *slow transition*, while every two frames of the third row is a *fast transition*)

Instance Occlusion (IO), more than 10% of the instance is occluded, manually labeled by annotators.

Illumination Variation (IV), illumination changes between previous and current frames. We use the Shade of Gray algorithm [52] of color constancy to calculate the correction matrix C_i between the current illumination and the standard illumination. Multiplying the original frame F_i and the correction matrix C_i can obtain the frame S_i under standard illumination. The gamma correction factor is 2.2 and the power is 6 in the correction matrix calculation. Subsequently, the cosine similarity between the vectors C_i and $O_i = [1, 1, 1]$ is calculated as the illumination standard i_i of the current frame: $i_i = \frac{C_i \bullet O_i}{\|C_i\| \times \|O_i\|}$. i_i is a quantization value of *Illumination Estimation (IE)*, and $i_i < 0.99$ means special illumination in current frame.The difference in absolute value between previous frame i_{i-1} and current frame i_i is Δi_i, $\Delta i_i > 0.0001$ means illumination variation in continuous frames.

Blur Variation (BV), quantization of the sharpness variation between previous and current frames. The variation of the Laplacian [53] is selected to calculate the blur degree. We first convert the current image F_i into a grayscale image G_i, then convolve G_i with a specific Laplacian kernel L, and calculate the variance of the response result v_i — this value is used as an index of sharpness. Images with $v_i < 100$ can be considered blurry; otherwise are clear. Besides, the difference in absolute value between v_{i-1} and v_i is Δv_i, while $\Delta v_i > 1.5$ means blur variation.

Scale Variation (SV), indicator for measuring changes in instance scales. The size of instance in current frame is $s_i = \sqrt{w_i h_i}$, and $s_i \notin [50, 750]$ will be considered as *Special Scale (SS)*. Δs_i is calculated by the difference of absolute value between s_{i-1} and s_i, and $\Delta s_i > 30$ signifies scale variation.

Ratio Variation (RV), indicator for characterizing the target deformation and rotation. The aspect ratio of instance in current frame is $r_i = \frac{h_i}{w_i}$, and $r_i \notin [\frac{1}{3}, 3]$ will be considered as

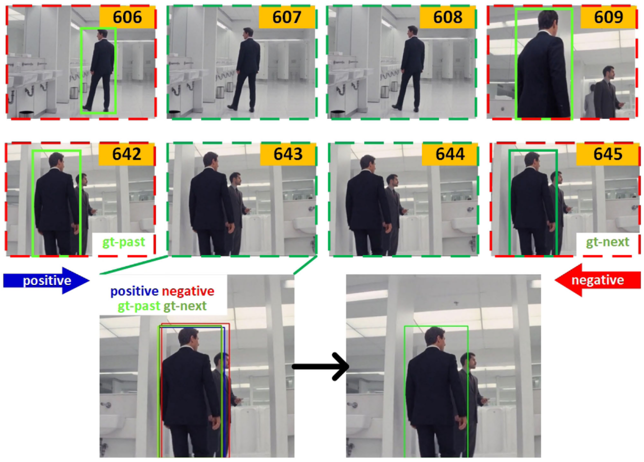
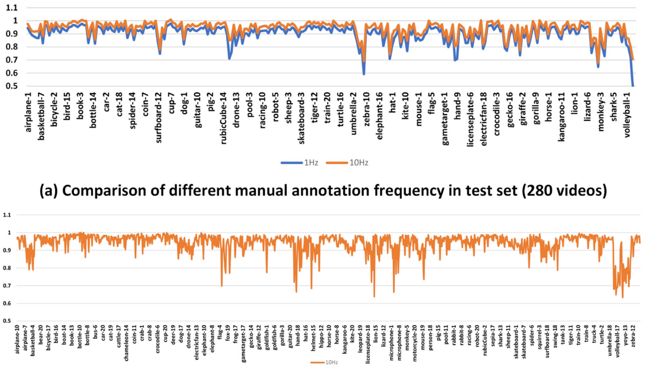


Fig. 13. Examples of automatic annotation.



(a) Comparison of different manual annotation frequency in test set (280 videos)

(b) The IoU value of all 1400 videos based on 10Hz manual annotation frequency

Fig. 14. The experimental result of the automatic annotation on LaSOT.

Special Ratio (SR). Same as the previous calculation process, $\Delta r_i > 0.2$ stands for ratio variation.

Algorithm 1: Framework of Generate the Automatic Annotation

```
Input: $P_{gt}$: previous mannually labeled bounding-box; $N_{gt}$:
       next mannually labeled bounding-box; $B_{pos}$: bounding-
       box generated in forward order; $B_{neg}$: bounding-box
       generated in backward order
Output: $B$: bounding-box of present frame
calculate $D_1 = DIoU(P_{gt}, N_{gt})$
calculate $D_2 = DIoU(B_{pos}, B_{neg})$
 /* Situation 1: a high value of $D_1$ indicates miniature movement, and the location can be directly calculated */
if $D_1 \geq \tau_1$ then
    $B = average(P_{gt}, N_{gt})$ return $B$
calculate $E_1 = Enclose(P_{gt}, N_{gt})$
 /* Situation 2: a high value of $D_2$ indicates normal movement, and this is the most common situation. We assume that the motion range of instance in intermediate frame does not exceed $E_1$ */
if $D_2 \geq \tau_2$ then
    if both $B_{pos}$ and $B_{neg}$ are enclosed by $E_1$ then
        $B = average(B_{pos}, B_{neg})$
    else if $B_{pos}$ or $B_{neg}$ is enclosed by $E_1$ then
        $B = B_{pos}$ or $B = B_{neg}$
    else if both $B_{pos}$ and $B_{neg}$ are outside $E_1$ then
        $B = average(P_{gt}, N_{gt})$
    return $B$
/* Situation 3: the situation does not belong to the above two conditions indicates rapid movement or shot-switching */
if presnet frame is the last two frame in a shot then
    $B = average(B_{pos}, P_{gt})$
else if presnet frame is the first two frame in a shot then
    $B = average(B_{neg}, N_{gt})$
else
    calculate $D_3 = DIoU(P_{gt}, B_{pos})$
    calculate $D_4 = DIoU(N_{gt}, B_{neg})$
    if $D_3 \geq D_4$ then
        $B = Intersection(E_1, B_{pos})$
    else
        $B = Intersection(E_1, B_{neg})$
return $B$
```

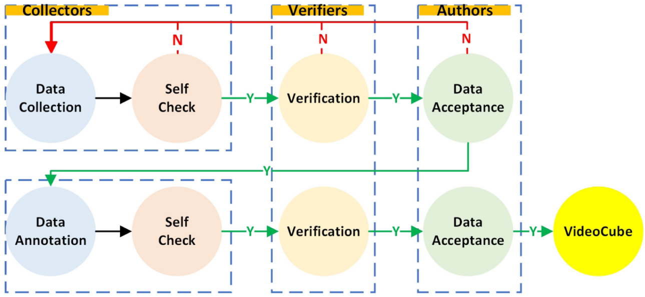


Fig. 15. The framework of data checkout process.

Fast Motion (FM), an index $d_i = \frac{\|c_i - c_{i-1}\|_2}{\sqrt{s_i s_{i-1}}}$ aims to measure the instance motion speed. The motion of object in current frame is d_i, where the c_i indicates the center of bounding box. Since d_i has reflected the dynamic relationship between F_i and F_{i-1}, it can be used to reflect the motion variation between two frames directly. $d_i > 0.2$ will be treated as fast motion.

Correlation Coefficient (CC), a metric used to measure the similarity between F_i and F_{i-1}. In this paper, we use the Pearson product-moment correlation coefficient(PPMCC) $p_i = \rho_{i,i-1} = \frac{\mathrm{cov}(F_i, F_{i-1})}{\sigma_{F_i}\sigma_{F_{i-1}}}$. The numerator calculates the covariance of the current frame F_i and the previous frame F_{i-1}, and the denominator is the product of the standard deviation. The correlation coefficient reflects the changes between consecutive frames and has been normalized; it can be used as an attribute index directly. $p_i > 0.8$ signifies the correlation between the continuous two frames is strong.

Twelve attributes can be divided into three types: filtering attributes, self attributes, and dynamic attributes. Instance absent (IA) and shot-cut (SC) are filtering attributes to remove frames that are unsuitable for metrics calculation in experiments. Instance occlusion (IO), illumination estimation (IE), special scale (SS), and special ratio (SR) are self attributes that only reflect the characteristics of the current frame rather than embody the dynamic variations. Blur variation (BV), illumination variation (IV), scale variation (SV), ratio variation (RV), fast motion (FM), and coefficient of correlation (CC) are dynamic attributes that contain the dynamic relationship between consecutive frames.

3.2 Evaluation System

The following two sections introduce the evaluation system and performance measurement of the GIT task. The evaluation system aims to judge the model's capabilities (such as accuracy and robustness) through a reasonable evaluation method. The performance measurement focuses on quantitatively mapping the model capabilities through scientific calculation to accomplish more in-depth analysis and sort the results via numerical values.

3.2.1 One-Pass Evaluation (OPE)

The evaluation protocol of OTB [1] benchmark has six categories: three normal processes, including one-pass evaluation (OPE), temporal robustness evaluation (TRE), spatial robustness evaluation (SRE), and three restart processes involving one-pass evaluation with restart (OPER), temporal robustness evaluation with restart (TRER), spatial robustness evaluation with restart (SRER). Among them, the OPE

TABLE 1
Comparison of VideoCube With Popular Single Object Tracking Benchmarks

Benchmark	Year	Videos	Min Frame	Mean Frame	Median Frame	Max Frame	Total Frame	Total Duration	Label Density	Attribute Classes (Absent)	Object Classes	Motion Modes	Scene Categories
OTB2013 [34]	2013	51	71	578	392	3872	29K	16.4m	30Hz	11(✗)	10	n/a	n/a
OTB2015 [1]	2015	100	71	590	393	3872	59K	32.8m	30Hz	11(✗)	16	n/a	n/a
TC-128 [41]	2015	129	71	429	365	3872	55K	30.7m	30Hz	11(✗)	27	n/a	n/a
NUS-PRO [42]	2015	365	146	371	300	5040	135K	75.2m	30Hz	n/a	8	n/a	n/a
UAV123 [43]	2016	123	109	915	882	3085	113K	75.2m	30Hz	12(✗)	9	n/a	n/a
VOT-2017 [4]	2017	60	41	356	293	1500	21K	11.9m	30Hz	n/a	24	n/a	n/a
Nfs [44]	2017	100	169	3830	2448	20665	383K	26.6m	240Hz	9(✗)	17	n/a	n/a
TrackingNet [2]	2018	30643	-	498	-	-	14M	141h	1Hz(30Hz)^{a}	15(✗)	27	n/a	n/a
GOT-10k [5]	2019	10000	29	149	101	1418	1.45M	40h	10Hz^{b}	6(✓)	563^{c}	87	n/a
UAV20L [43]	2016	20	1717	2934	2626	5527	59K	32.6m	30Hz	12(✗)	5	n/a	n/a
OxUvA [46]	2018	366	900	4320	2628	37740	1.55M	14.4h	1Hz^{d}	(✓)^{e}	22	n/a	n/a
LaSOT [3]	2020	1550	1000	2502	2145	11397	3.87M	35.8h	30Hz	14(✓)	85	n/a	n/a
VideoCube	2020	500	**4008**	**14920**	**14162**	**29834**	**7.46M**	**69.1h**	10Hz(30Hz)^{f}	12(✓)	**9(89)**^{g}	**61**	**8(55)**^{h}

VideoCube is superior to existing datasets in multiple dimensions, including scale, label density, and content richness (object classes, motion modes, scene categories). Note: (a) TrackingNet performs manual annotation per second and uses the DCF [47] algorithm to automatically label the remaining frames to accomplish dense labeling with 30Hz frequency. (b) GOT-10k extracts 1.45 million images from more than 40h videos at 10FPS and manually annotates each frame. (c) The object classes in GOT-10k are finely divided based on WordNet [45]. For example, the border collie is an independent category, rather than being divided into dogs. (d) OxUvA believes that the manual labeling frequency of 1 Hz is sufficient for trackers, thus only offering annotation once per second. (e) OxUvA only performs additional annotation about target absence but ignores other challenging attributes. (f) VideoCube combines manual and automatic annotation similar to TrackingNet but increases the manual label frequency to 10Hz due to frequent scene switching in videos, and uses PrDiMP [48] to complete 30Hz dense annotation. (g) VideoCube uses WordNet as the semantic framework to divide the video objects into 9 categories and 89 sub-categories. (h) Given WordNet's limited ability to classify unique scenes, VideoCube uses WordNet as the backbone and references FrameNet [49] and ConceptNet [50] to divide scenes into 8 categories and 55 sub-categories.

method is defined as using the ground-truth in the first frame to initialize the model and continuously locate the target in subsequent frames. Subsequent tracking-based visual tasks (i.e., short-term and long-term tracking) are only distinguished in assignment settings but maintain the OPE method as the evaluation system.

Numerous benchmarks listed in Table 1 except two short-term tracking benchmarks (OTB and VOT) only retain the OPE mode but discard the restart mechanism. However, the restart proposed by OTB does not perform real-time supervision but generates the OPER results based on a series of existing experimental results generated by the TRE method. Although VOT can complete a fail-detection in the algorithm running process, the re-initialization is performed directly without any design for selecting the restart frame.

3.2.2 One-Pass Evaluation With Restart Mechanism (R-OPE)

The restart mechanism is essential in evaluating the GIT task for the following reasons: (1) Videos in VideoCube have a longer average frame and include multiple challenging characteristics like shot-switching and scene-transferring. Thus, models are prone to fail in locating instances and cannot be reinitialized. (2) The count of restarts can be quantified as an indicator to measure the algorithm's robustness. A similar restart mechanism has been studied on monkeys by neuroscientists [54], [55]. They replace the target P with interference N during the rapid eye movements of monkeys. After several repetitions, the observation of activities in the temporal cortex of monkeys indicates that the monkey has confused P and N. Some online update algorithms continue learning the apparent characteristics of the instance during the tracking process. However, challenges like lens-switching mean the algorithm needs to expand the search range to relocate the instance (like rapid eye movement). Re-location may misidentify the interference as an instance and update the wrong sample. This situation is caused by the wrong instance updating rather than weak learning ability. Therefore, VideoCube includes two evaluation systems: traditional OPE and *OPE with restart mechanism (R-OPE)*.

The foundation of the R-OPE mechanism is selecting restart frames. The selection process follows two principles: (1) The restart frame is manually annotated rather than automatically generated to ensure the label quality. (2) Rich instance features are contained in the restart frame to provide enough information for re-initialization. We select the YOLACT++ [56] algorithm to segment each manually labeled bounding box, delete the background, match the remaining instance with the clear-cut query in the first frame, and finally determine frames with matching points exceeding a certain threshold as restart frames. According to statistics, 17.2% of frames in VideoCube satisfy the filter conditions.

Example of R-OPE mechanism is shown in Fig. 16. The first row illustrates the traditional OPE mechanism. The tracker is initialized in the first frame, in which the algorithm result (blue) and ground-truth (red) coincide. In the following tracking process, the IoU value of the algorithm result (blue) and ground-truth (red) is less than a threshold (usually 0.5) in the #130, which indicates a failure. Since the OPE mechanism does not detect failure, the continued failure causes subsequent frames to be wasted. The second row

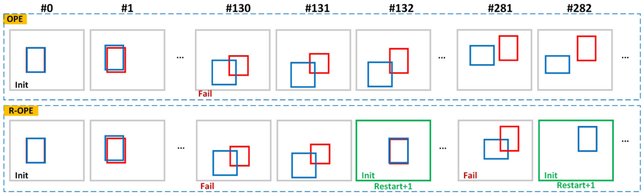
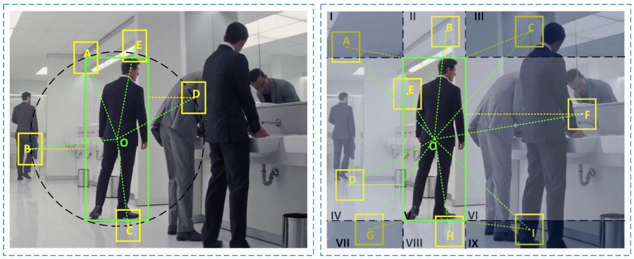


Fig. 16. Comparison of the two evaluation mechanisms. The first row illustrates traditional OPE mechanism, and the second row illustrates R-OPE mechanism with failure detection and tracker restart.



(a) Insufficiency of existing precision indicators

(b) The computation process of the normalized precision

Fig. 17. The counterexample of traditional precision metrics and the computation process of the normalized precision (N-PRE). (*a*) Insufficiency of existing precision indicators. Common sense infers that algorithm A has the highest accuracy while B performs worst. However, the traditional precision (TRE) indicator results consider A, B, C, D have the same precision and are better than E. (*b*) The computation process of the normalized precision (N-PRE). The ground-truth bounding box divides the screen into nine areas (*I* to *IX*). Point E falls into area *V* (ground-truth); the distance between E and the O point is considered the original precision value of tracker E. For other trackers that fall into eight external areas, the original precision value is the sum of two parts. The first part is the distance between the center point and O (shown as the green dashed line); the second part is the penalty item calculated by the shortest distance between the center point and the edge of the ground-truth box (shown as the yellow dashed line). To exclude the influence of instance size and frame resolution, we select the maximum value of all screen points to normalize the result.

is the R-OPE mechanism with failure detection and tracker restart. The green frame indicates an appropriate restart point. After the tracking failure is detected at #130, the algorithm will be re-initialized at the nearest restart point (#132), and subsequent sequences will continue to participate in the evaluation. When the tracking failure occurs in #281, the algorithm will be restarted at #282.

3.3 Evaluation Metrics

Similar to the metrics used by most SOT benchmarks [1], [3], [5], [46], we first utilize the precision plot and the success plot to measure the performance of the algorithms for OPE and R-OPE mechanisms.

3.3.1 Precision Plot

Tradition precision (PRE) measures the center distance between the predicted result p^t and the ground-truth b^t in pixels. Calculating the proportion of frames whose distance is less than the specified threshold and drawing the statistical results based on different thresholds into a curve generates the precision plot. Typically, trackers are ranked on 20 pixels [1], [3]. However, the object scale is influenced by target size and image resolution but ignored by the original PRE score. Thus, two new benchmarks [2], [3] adopt the ground-truth scale (width and height) to normalize the center distance. Specifically, the height difference and width difference between two center points are divided by the ground-truth shape before calculating the distance. This operation solves the target scale influence on PRE calculation but is still not comprehensive enough.

Fig. 17a presents a counterexample. The green rectangle represents the ground-truth, where point O denotes the center point. Assume that five yellow rectangular boxes show the prediction results of the five algorithms. To eliminate the influence of other factors, here assumes the prediction results have only position differences. OA, OB, OC, and OD are the same, while OE is slightly larger. The precision scores of tracker A, B, C, D based on two existing metrics (directly using the center point distance or only using the current ground-truth size for normalization) are the same, while tracker E is worse. Nevertheless, from Fig. 17a, we can directly judge that A and E perform better than B. The calculation results are contrary to common sense because the target aspect ratio affects accuracy but is ignored by existing metrics. For non-square bounding boxes, only the center point distance cannot quantify the tracking accuracy accurately.

To deal with the above problem, we propose a new precision metric N-PRE. Explicitly, we select the center distance as the original precision if the tracker center point falls into the ground-truth rectangle. Algorithms with a predicted center outside the ground-truth rectangle will also calculate the shortest distance between its center and the ground-truth edge. As shown in Fig. 17b, the original precision value of tracker E is OE, while other trackers are calculated by two parts (center distance represented by the green dashed line, and the penalty item represented by the yellow dashed line). Subsequently, we quantify the original precision value to the [0, 1] interval; 0 represents the tracker center point is O, while 1 represents the score of the farthest point in the current frame (upper right point). In Fig. 17a, the performance of tracker A evaluated via N-PRE is the best while tracker B is the worst. It is consistent with reality.

3.3.2 Success Plot

To get the success rate (SR), we first calculate the Intersection over Union (IoU) of the predicted result p^t and the ground-truth b^t. Frames with an overlap rate greater than a specified threshold are defined as successful tracking, and the SR measures the percentage of successfully tracked frames under different overlap thresholds. The statistical results based on different thresholds create the success plot. Besides, we implement two more success scores based on Generalized IoU (GIoU [57]) and Distance IoU (DIoU [58]), aiming to provide a comprehensive scientific evaluation.

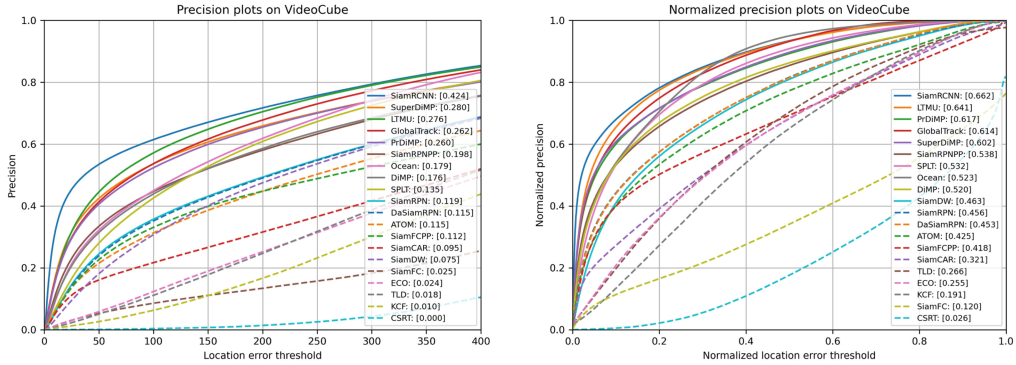
3.3.3 Robustness

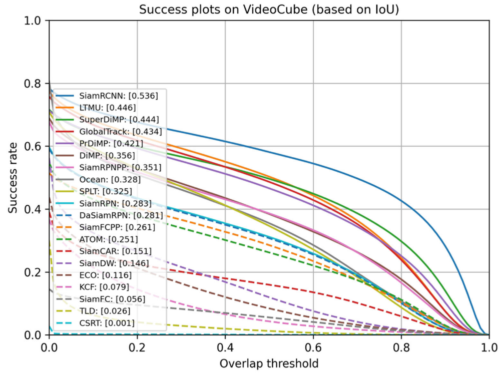
For the R-OPE mechanism, we propose a new evaluation indicator to evaluate robustness. Specifically, we define robustness as $R = \frac{1}{N}\sum_{i=1}^{N}[S(\frac{1}{\rho_i})(1-\frac{I_i}{R_i})]$. N represents the number of videos participating in the evaluation, ρ_i indicates the correlation coefficient of the ith video, R_i means the total number of restart frames selected for this video, and I_i denotes the number of restarts of the tracker.

4 Experiments

We accomplish extensive experiments in this section and divide them into two parts:

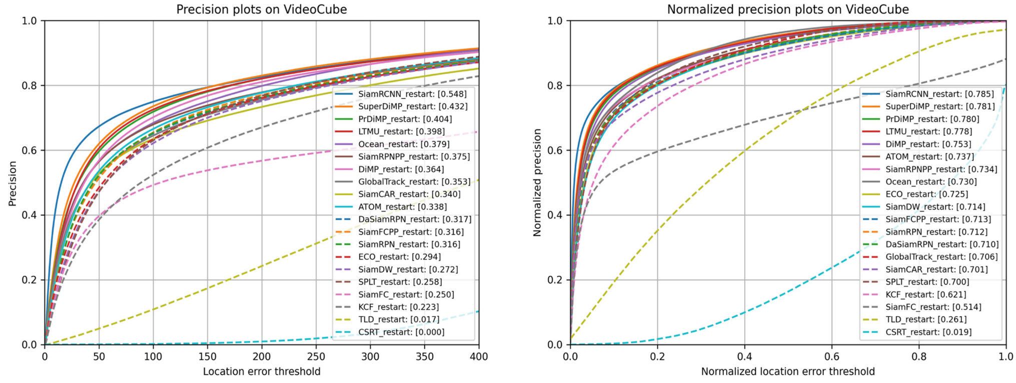
Standard Experiments. We select 20 algorithms (Ocean [59], SiamRCNN [60], SuperDiMP [48], LTMU [61], PrDiMP [48], SiamCAR [62], SiamFC++ [63], SiamDW [64], GlobalTrack [65], DiMP [66], SPLT [67], SiamRPN++ [68], ATOM [69], DaSiamRPN [70], SiamRPN [30], ECO [27], SiamFC [28],

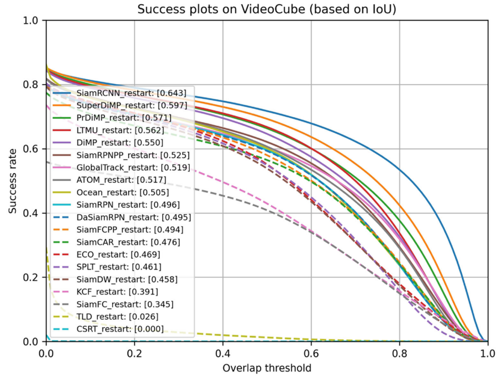




Trackers	Venue	PRE (20 pixels)	N-PRE (percent in area V)	SR-IoU (mean)	SR-GIoU (mean)	SR-DIoU (mean)	FPS
Ocean	ECCV'20	0.179	0.523	0.328	0.316	0.312	11.665
SiamRCNN	CVPR'20	**0.424**	**0.662**	**0.536**	**0.529**	**0.524**	2.39
SuperDiMP	CVPR'20	**0.28**	0.602	**0.444**	**0.436**	**0.433**	6.423
PrDiMP	CVPR'20	0.26	**0.617**	0.421	0.412	0.408	5.791
LTMU	CVPR'20	**0.276**	**0.641**	**0.446**	**0.435**	**0.433**	2.512
SiamCAR	CVPR'20	0.095	0.321	0.151	0.144	0.142	11.063
SiamFCPP	AAAI'20	0.112	0.418	0.261	0.251	0.251	**79.067**
GlobalTrack	AAAI'20	0.262	0.688	0.434	0.424	0.42	3.712
DiMP	ICCV'19	0.176	0.52	0.356	0.346	0.342	6.252
SPLT	ICCV'19	0.135	0.532	0.325	0.311	0.309	11.436
SiamDW	CVPR'19	0.075	0.463	0.146	0.132	0.134	8.766
SiamRPNPP	CVPR'19	0.198	0.538	0.351	0.34	0.339	6.363
ATOM	CVPR'19	0.115	0.425	0.251	0.242	0.238	6.963
DaSiamRPN	CVPR'18	0.115	0.453	0.281	0.271	0.268	24.37
SiamRPN	CVPR'18	0.119	0.456	0.283	0.274	0.271	44.454
CSRT	IJCV'18	0	0.013	0.001	0.001	0.001	**79.831**
ECO	CVPR'17	0.024	0.255	0.116	0.102	0.101	7.176
SiamFC	ECCV'16	0.025	0.12	0.056	0.053	0.053	47.928
KCF	TPAMI'15	0.01	0.026	0.079	0.062	0.062	**162.962**
TLD	TPAMI'11	0.018	0.266	0.026	0.022	0.019	5.99

Fig. 18. Standard experiments in OPE mechanism, evaluated by precision (PRE) plot, N-PRE plot, and success plot. The red, blue, and green in the tables represent the first, second, and third placed algorithms of each indicator.



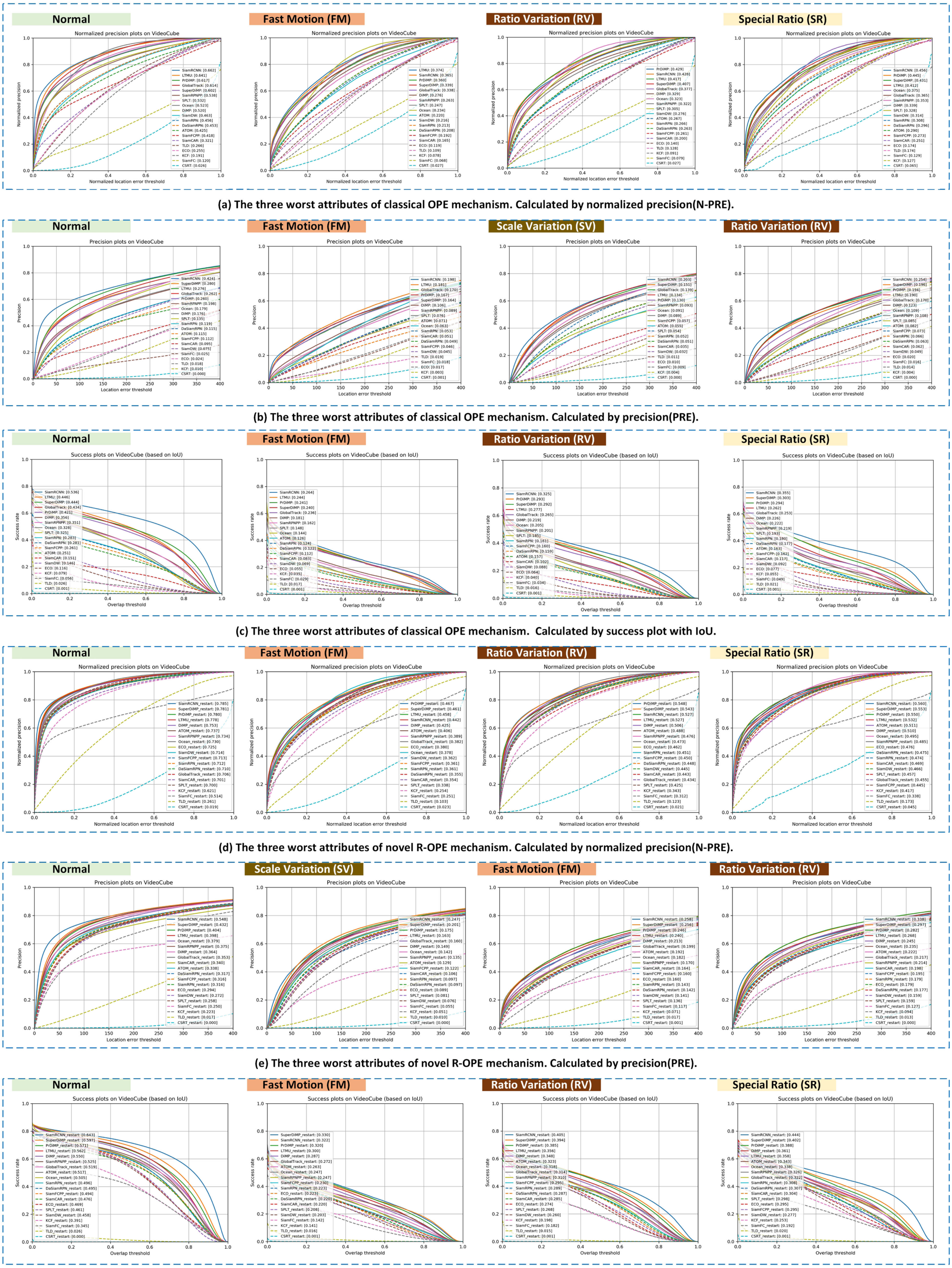


Trackers	Venue	PRE (20 pixels)	N-PRE (percent in area V)	SR-IoU (mean)	SR-GIoU (mean)	SR-DIoU (mean)	Robust
Ocean	ECCV'20	0.379	0.73	0.505	0.493	0.495	0.7349
SiamRCNN	CVPR'20	**0.548**	**0.785**	**0.643**	**0.637**	**0.635**	**0.7426**
SuperDiMP	CVPR'20	**0.432**	**0.781**	**0.597**	**0.59**	**0.588**	**0.7405**
PrDiMP	CVPR'20	**0.404**	**0.78**	**0.571**	**0.563**	**0.561**	0.7397
LTMU	CVPR'20	0.398	0.778	0.562	0.553	0.553	0.7397
SiamCAR	CVPR'20	0.34	0.701	0.476	0.464	0.466	0.7332
SiamFCPP	AAAI'20	0.316	0.713	0.494	0.481	0.484	0.7338
GlobalTrack	AAAI'20	0.353	0.706	0.519	0.51	0.508	**0.7404**
DiMP	ICCV'19	0.364	0.753	0.55	0.54	0.54	0.7375
SPLT	ICCV'19	0.258	0.7	0.461	0.447	0.448	0.7321
SiamDW	CVPR'19	0.272	0.714	0.458	0.441	0.446	0.7313
SiamRPNPP	CVPR'19	0.375	0.734	0.525	0.516	0.515	0.7365
ATOM	CVPR'19	0.338	0.737	0.517	0.507	0.506	0.7355
DaSiamRPN	CVPR'18	0.317	0.71	0.495	0.484	0.484	0.7335
SiamRPN	CVPR'18	0.316	0.712	0.496	0.485	0.486	0.7336
CSRT	IJCV'18	0	0.019	0	0	0	0.6848
ECO	CVPR'17	0.294	0.725	0.469	0.453	0.457	0.7318
SiamFC	ECCV'16	0.25	0.514	0.345	0.334	0.338	0.7254
KCF	TPAMI'15	0.223	0.621	0.391	0.373	0.376	0.7224
TLD	TPAMI'11	0.017	0.261	0.026	0.022	0.019	0.6868

Fig. 19. Standard experiments in R-OPE mechanism, evaluated by precision (PRE) plot, N-PRE plot, and success plot. The red, blue, and green in the tables represent the first, second, and third placed algorithms of each indicator.

TLD [71], CSRT [72], KCF [23]) as baselines and conduct experiments under the OPE and R-OPE mechanisms. All algorithms are fully evaluated under two mechanisms to generate the precision plot and success plot.

Eye Tracking Experiments. We apply an eye tracker machine to record and quantify the human visual tracking ability. The intelligence level of trackers can be measured by comparing human capacity with algorithm tracking results.



(a) The three worst attributes of classical OPE mechanism. Calculated by normalized precision(N-PRE).

(b) The three worst attributes of classical OPE mechanism. Calculated by precision(PRE).

(c) The three worst attributes of classical OPE mechanism. Calculated by success plot with IoU.

(d) The three worst attributes of novel R-OPE mechanism. Calculated by normalized precision(N-PRE).

(e) The three worst attributes of novel R-OPE mechanism. Calculated by precision(PRE).

(f) The three worst attributes of novel R-OPE mechanism. Calculated by success plot with IoU.

Fig. 20. The attribute performance. (*a*) to (*c*) illustrates the performance of the three worst attributes in classical OPE mechanism by different evaluation metrics. (*d*) to (*f*) illustrates the performance of the three worst attributes in R-OPE mechanism by different evaluation metrics.

4.1 Standard Experiments

Twenty trackers are selected as baseline models and evaluated on VideoCube. Given that most algorithms do not determine the instance absent, we first remove the frames that exclude the tracked instance. Besides, frames in the transition stage may include superimposed instances. To ensure the accuracy of the evaluation, we remove the transition frames as well.

Trackers	Venue	PRE (20 pixels)	N-PRE (percent in area V)
Ocean	ECCV'20	0.256	0.476
SiamRCNN	CVPR'20	0.551	0.71
SuperDiMP	CVPR'20	0.398	0.617
PrDiMP	CVPR'20	0.354	0.59
LTMU	CVPR'20	0.421	0.662
SiamCAR	CVPR'20	0.142	0.4
SiamFCPP	AAAI'20	0.153	0.412
GlobalTrack	AAAI'20	0.405	0.687
DiMP	ICCV'19	0.26	0.487
SPLT	ICCV'19	0.158	0.501
SiamDW	CVPR'19	0.106	0.431
SiamRPNPP	CVPR'19	0.262	0.521
ATOM	CVPR'19	0.151	0.408
DaSiamRPN	CVPR'18	0.136	0.39
SiamRPN	CVPR'18	0.132	0.371
CSRT	IJCV'18	0.001	0.116
ECO	CVPR'17	0.028	0.208
SiamFC	ECCV'16	0.044	0.143
KCF	TPAMI'15	0.005	0.141
TLD	TPAMI'11	0.019	0.293
Turing_15	Human	0.377	0.85
Turing_20	Human	0.243	0.805
Turing_30	Human	0.203	0.778

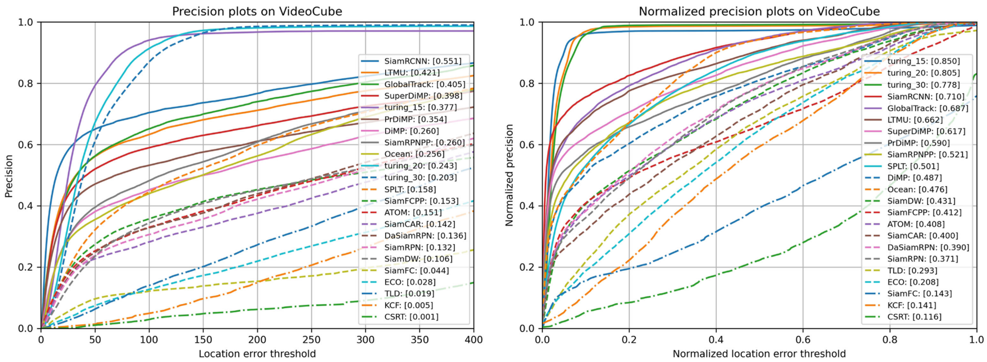


Fig. 21. Eye-tracking experiments in OPE mechanism, evaluated by precision (PRE) plot and N-PRE plot. The red, blue, and green in the tables represent the first, second, and third placed algorithms of each indicator.

4.1.1 Overall Performance

Figs. 18 and 19 present the overall performance of trackers in OPE and R-OPE mechanisms. The scores and rankings of algorithms under these two mechanisms are pretty distinct, confirming that the two evaluation mechanisms' focuses are different. For evaluation results in OPE (Fig. 18), the algorithm scores are low since the VideoCube allows lens switching and scene transferring, causing the jump change of the target position in consecutive frames. Most algorithms strongly depend on continuous motion assumption and usually use local search to locate the target, thus performing worse when the position variation occurs. The R-OPE mechanism restarts algorithms at the next restart point after detecting the failure (Fig. 19). Its precision plot and success plot focus on evaluating the local-search ability, while the robustness score obtained via quantifying the number of restarts reflects the global-search ability.

4.1.2 Attribute Performance

VideoCube selects twelve attributes to describe the challenges in the GIT task and divides them into three categories: filtering attributes, self attributes, and dynamic attributes. We provide twelve attribute labels for each frame to fully capture the difficulty factors. The detailed results are demonstrated in Fig. 20. It is clear that compared with other attributes, *fast motion (FM)*, *ratio variation (RV)*, *special ratio (SR)*, and *scale variation (SV)* challenge the performance of trackers.

4.2 Eye Tracking Experiments

Unlike traditional visual tracking experiments that only evaluate algorithms with performance rather than intelligence, we design an eye-tracking experiment to judge human visual tracking ability and measure machine intelligence via comparison.

Ten videos with different difficulty, duration, instance types, space classes, motion modes are played to the subject at three speeds (15FPS, 20FPS, and 30FPS). We select 15 human subjects to track the test video at three rates. We have obtained the approvals of all the human participants. Every participant has signed an informed consent form before the experiment. First, each subject should calibrate the eye tracker machine to ensure that the instrument can accurately detect the sightline. Second, the test video appears in the screen center, and the subject should focus on the target in the first frame, then press the play button. After that, the subject needs to concentrate on the target and maintain tracking accuracy. The subject has a rest time to relieve visual fatigue between two videos. The eye tracker machine records the eye movement of subjects, and the focus of sight is used to calculate the precision score and generate precision plots.

Fig. 22 illustrates the detailed process of eye-tracking experiments, which consists of three steps. (1) The subject calibrates the eye tracker machine (Tobii Eye Tracker) to ensure that the instrument can accurately detect the sightline. (2) A TEST video appears in the screen center. The subject should focus on the target in the first frame, press the play button, and concentrate on maintaining tracking accuracy. TEST video aims to help subjects familiarize themselves with the test process. (3) The subject begins the formal experiment by tracking six different videos. A break between two videos is needed to ensure the effectiveness of the experiment.

Fig. 21 presents the precision plots of humans and 20 trackers in OPE mechanisms.Turing_15, Turing_20, and Turing_30 represent human scores at 15FPS, 20FPS, and 30FPS, respectively. We can draw the following conclusions through comparison: (1) The calculation methods and sequencing principles of traditional precision (PRE) scores have multiple problems. PRE measures the center distance between the predicted result and the ground-truth in pixels, but ignores the impact of target size and video resolution (for detailed analysis, please refer to the methods chapter). This makes the ranking threshold with 20 pixels unreasonable. In the precision plot of Fig. 21, human performance is far lower than algorithms, contrary to our common sense. Since the deviation of the eye tracker machine may exceed 20 pixels in several situations, 20 pixels are too strict by

Exp Number	20FPS	30FPS		20FPS		15FPS	
1	TEST	A	J	B	E	C	H
2	TEST	B	C	J	D	E	F
3	TEST	D	E	F	I	A	H
4	TEST	F	G	C	A	D	B
5	TEST	H	I	G	E	A	J
6	TEST	C	J	A	H	G	I
7	TEST	A	B	F	J	G	E
8	TEST	D	I	G	H	C	B
9	TEST	E	J	D	C	F	I
10	TEST	F	A	B	E	J	H
11	TEST	H	I	B	C	D	G
12	TEST	G	B	H	I	E	C
13	TEST	C	G	A	D	B	F
14	TEST	D	H	F	J	A	I
15	TEST	E	F	G	I	D	J

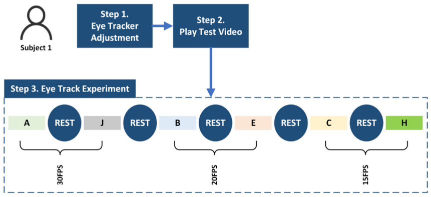


Fig. 22. The process of eye-tracking experiment. Ten videos (A-J) with different difficulty, duration, instance types, space classes, motion modes are played to the subject at three speeds (15FPS, 20FPS, and 30FPS). Fifteen subjects track the test videos at three rates.

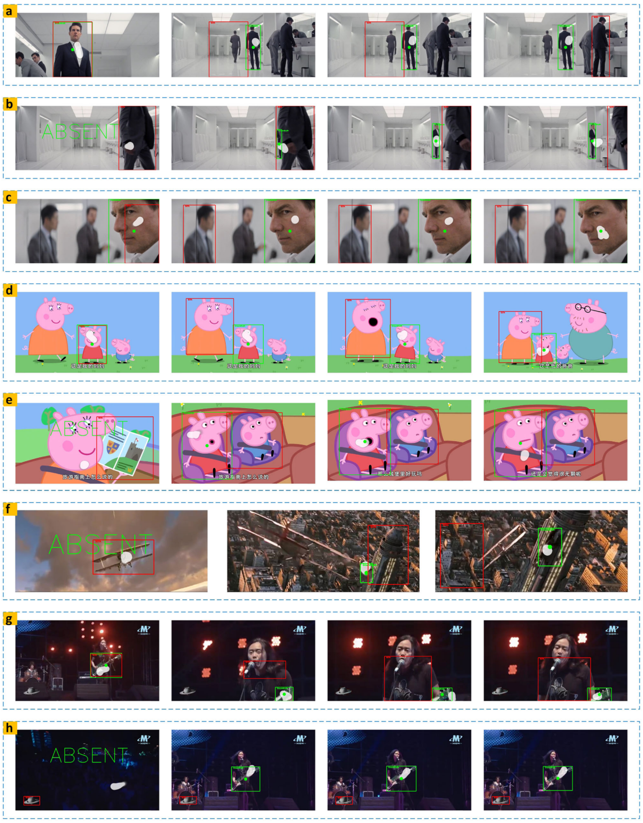


Fig. 23. Some examples of human visual tracking ability better than SOTA algorithm. (*a*, *e*, and *f*) When the *transition* occurs, the human can locate the target immediately, while the algorithm drifts to a similar object. (*b*) When the *obstruction* is removed, the human can locate the target immediately, but the algorithm keeps tracking the obstruction. (*c* and *d*) When *same-category objects* appear on the screen, people can distinguish between the target and similar items, but the algorithm fails. (*g* and *h*) The emergence of *transitions* and the *complex illumination environment* makes the algorithm unable to locate the target, but humans can quickly identify the target (guitar) through auxiliary information (the guitarist).

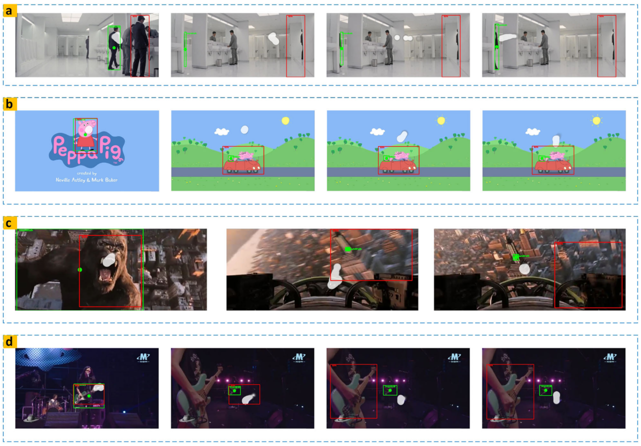


Fig. 24. Some examples where both the human and the SOTA algorithm fail. (*a*) The *transition* causes the jumps of target position, and the *occlusion* factor in the new scene makes it impossible for both humans and algorithms to locate the target quickly. (*b*) The *transition*, the presence of *interference* and the *tiny size* make it challenging to locate the target. (*c*) The *transition*, the frame *blur* due to *fast movement* and the *tiny size* make it challenging to locate the target. (*d*) The *transition*, the *tiny size* and the *complex illumination environment* make it challenging to locate the target.

comparing with the image resolution and target size of videos in VideoCube. (2) The normalized precision plot shows that the human visual tracking ability is worse than tracking algorithms for strict precision requirements. The reason may be the deviation of the eye tracker machine and the human attention (for person target, subjects prone to focus on the head instead of the torso). When the accuracy requirements are moderately reduced, the human visual ability will quickly exceed algorithms and remain stable.

Besides, Fig. 23 explains how the human eye outperforms the SOTA algorithms. Humans can quickly locate the target in challenging factors such as transitions, similar objects, occlusion, and complex illumination occur, while the algorithms always fail or drift to similar instances. The experiment presents that algorithms need to enhance the robustness in challenging environments to achieve human-like tracking. Fig. 24 presents cases where human eyes and algorithms fail, indicating that some extreme environments challenge both humans and algorithms to accomplish target tracking.

5 Conclusion

To help trackers locate the target more like humans, we analyze the fundamentals of measuring the intelligence level and summarize the limitations of existing benchmarks. In this paper, we (1) propose the GIT task to explicitly model the human visual tracking ability, (2) build the VideoCube benchmark to create a challenging experimental environment close to the real world, and (3) finally design a scientific evaluation procedure to measure the tracking performance of humans and machines. The experimental results show that there is still a definite gap between trackers and humans. Still, we believe the general online platform treats human tracking capabilities as a baseline to evaluate the machine intelligence level, guiding the research to accomplish human-like trackers in the future.

References

[1] Y. Wu, J. Lim, and M.-H. Yang, “Object tracking benchmark,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[2] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, “TrackingNet: A large-scale dataset and benchmark for object tracking in the wild,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 310–327.

[3] H. Fan *et al.*, “LASOT: A high-quality large-scale single object tracking benchmark,” *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 439–461, 2021.

[4] M. Kristan *et al.*, “The visual object tracking vot2017 challenge results,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 1949–1972.

[5] L. Huang, X. Zhao, and K. Huang, “GOT-10K: A large high-diversity benchmark for generic object tracking in the wild,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[6] L. Hyvärinen, R. Walthes, N. Jacob, K. N. Chaplin, and M. Leonhardt, “Current understanding of what infants see,” *Curr. Ophthalmol. Rep.*, vol. 2, no. 4, pp. 142–149, 2014. [Online]. Available: https://europepmc.org/articles/PMC4243010

[7] P. Lennie and V. H. Sb, in *Visual Impairments: Determining Eligibility for Social Security Benefits*. Washington, DC, USA: Nat. Acad. Press, 2002.

[8] S. M. Marvastizadeh, L. Cheng, H. Ghaneiyakhdan, and S. Kasaei, “Deep learning for visual tracking: A comprehensive survey,” *IEEE Trans. Intell. Transp. Syst.*, 2019.

[9] G. Ciaparrone, F. L. Sanchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, “Deep learning in video multi-object tracking: A survey,” *Neurocomputing*, vol. 381, pp. 61–88, 2019.

[10] A. Esteva *et al.*, "Deep learning-enabled medical computer vision," *NPJ Digit. Med.*, vol. 4, no. 1, 2021, Art. no. 5.

[11] H. Dankert, L. Wang, E. D. Hoopfer, D. J. Anderson, and P. Perona, "Automated monitoring and analysis of social behavior in drosophila," *Nature Methods*, vol. 6, no. 4, pp. 297–303, 2009.

[12] A. Weissbrod *et al.*, "Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment." *Nature Commun.*, vol. 4, no. 1, 2013, Art. no. 2018.

[13] A. Mathis *et al.*, "Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning," *Nature Neurosci.*, vol. 21, no. 9, pp. 1281–1289, 2018.

[14] K. Wei and K. P. Kording, "Behavioral tracking gets real," *Nature Neurosci.*, vol. 21, no. 9, pp. 1146–1147, 2018.

[15] V. Ulman *et al.*, "An objective comparison of cell tracking algorithms," *Nat. Methods*, vol. 14, no. 12, pp. 1141–1152, 2017.

[16] V. M. K. Namboodiri *et al.*, "Single-cell activity tracking reveals that orbitofrontal neurons acquire and maintain a long-term memory to guide behavioral adaptation," *Nature Neurosci.*, vol. 22, no. 7, pp. 1110–1121, 2019.

[17] C. J. Payton and R. Bartlett, *Biomechanical Evaluation of Movement in Sport and Exercise: The British Association of Sport and Exercise Sciences Guide*, Evanston, IL, USA: Routledge, 2007.

[18] J. Dupeyroux, J. R. Serres, and S. Viollet, "Antbot: A six-legged walking robot able to home like desert ants in outdoor environments," *Sci. Robot.*, vol. 4, no. 27, 2019, Art. no. eaau0307.

[19] A. M. Turing, "Computing machinery and intelligence," in *Parsing the Turing Test*. Dordrecht, The Netherlands: Springer, 2009, pp. 23–65

[20] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[21] N. Brown and T. Sandholm, "Superhuman ai for heads-up no-limit poker: Libratus beats top professionals," *Science*, vol. 359, no. 6374, pp. 418–424, 2018.

[22] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. H. Heras, and G. G. d. Polavieja, "idtracker.ai: Tracking all individuals in small or large collectives of unmarked animals," *Nature Methods*, vol. 16, no. 2, pp. 179–182, 2019.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[24] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2544–2550.

[25] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf. Nottingham*, 2014. [Online]. Available: http://www.bmva.org/bmvc/2014/papers/paper038/index.html

[26] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4310–4318.

[27] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6931–6939.

[28] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1606.09549

[29] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1420–1429.

[30] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8971–8980.

[31] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1144–1152.

[32] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1349–1358.

[33] T. Yang and A. B. Chan, "Recurrent filter learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 2010–2019.

[34] Y. Wu, J. Lim, and M. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2411–2418.

[35] M. Kristan *et al.*, "The visual object tracking vot2013 challenge results," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2013, pp. 98–111.

[36] M. Kristan *et al.*, "The visual object tracking VOT2014 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 191–217.

[37] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2015, pp. 1–23.

[38] M. Kristan *et al.*, "The visual object tracking VOT2016 challenge results," 2016. [Online]. Available: http://www.springer.com/gp/book/9783319488806

[39] M. Kristan *et al.*, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 3–53.

[40] M. Kristan *et al.*, "The seventh visual object tracking VOT2019 challenge results," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2019, pp. 2206–2241.

[41] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[42] A. Li, M. Lin, Y. Wu, M.-H. Yang, and S. Yan, "NUS-PRO: A new visual tracking challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 335–349, Feb. 2016.

[43] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer, 2016, pp. 445–461.

[44] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1134–1143.

[45] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[46] J. Valmadre *et al.*, "Long-term tracking in the wild: A benchmark," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 692–707.

[47] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "DCFNet: Discriminant correlation filters network for visual tracking," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1704.04057

[48] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7181–7190.

[49] N. Schneider and C. Wooters, "The NLTK framenet API: designing for discoverability with a rich linguistic resource," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1703.07438

[50] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1612.03975

[51] D. Bordwell and K. Thompson, *Film Art: An Introduction*. New York, NY, USA: McGraw-Hill, 2011.

[52] G. D. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Proc. 12th Color Imag. Conf.*, 2004, pp. 37–41.

[53] J. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia, "Diatom autofocusing in brightfield microscopy: A comparative study," in *Proc. 15th Int. Conf. Pattern Recognit.*, 2000, pp. 314–317.

[54] N. Li and J. J. DiCarlo, "Unsupervised natural experience rapidly alters invariant object representation in visual cortex," *Science*, vol. 321, no. 5895, pp. 1502–1507, 2008.

[55] D. D. Cox, P. Meier, N. Oertelt, and J. J. DiCarlo, "'breaking' position-invariant object recognition," *Nature Neurosci.*, vol. 8, no. 9, pp. 1145–1147, 2005.

[56] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9157–9166.

[57] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.

[58] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 12993–13000, 2020.

[59] Z. Zhang and H. Peng, "Ocean: Object-aware anchor-free tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 771–787.

[60] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6578–6588.

[61] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, "High-performance long-term tracking with meta-updater," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6298–6307.

[62] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6269–6277.

[63] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 12549–12556, 2020.

[64] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4591–4600.

[65] L. Huang, X. Zhao, and K. Huang, "Globaltrack: A simple and strong baseline for long-term tracking," 2019, *arXiv: 1912.08531*.

[66] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1904.07220

[67] B. Yan, H. Zhao, D. Wang, H. Lu, and X. Yang, "'skimming-perusal' tracking: A framework for real-time and robust long-term tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2385–2393.

[68] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN+ +: Evolution of siamese visual tracking with very deep networks," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1812.11703

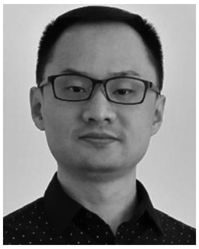
[69] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: accurate tracking by overlap maximization," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1811.07628

[70] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1808.06048

[71] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[72] A. Lukeźiǎź, T. Vojiř, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter tracker with channel and spatial reliability," *Int. J. Comput. Vis.*, vol. 126, no. 7, pp. 671–688, 2018.

Shiyu Hu received the BSc degree from the Beijing Institute of Technology and the MSc degree from the University of Hong Kong. In September 2019, she joined the Institute of Automation, Chinese Academy of Sciences, where she is currently working toward the Doctoral degree. Her current research interests include pattern recognition, computer vision, and machine learning.



Xin Zhao received the PhD degree from the University of Science and Technology of China. He is currently an associate professor with the Institute of Automation, Chinese Academy of Sciences. His current research interests include pattern recognition, computer vision, and machine learning. He was the recipient of the International Association of Pattern Recognition Best Student Paper Award at ACPR 2011. He was the recipient of the 2nd place entry of COCO Panoptic Challenge at ECCV 2018.

Lianghua Huang received the PhD degree from the Institute of Automation, Chinese Academy of Sciences. He has authored or coauthored 17 papers in the areas of computer vision and pattern recognition at international journals and conferences, including TPAMI, CVPR, ICCV, AAAI, ACMMM, TMM, TIP, TCYB, and ICME. His current research interests include representation learning and generative modeling in computer vision. He was the recipient of the 1st place entry of VizWiz VQA challenge at CVPR 2021.

Kaiqi Huang received the BSc and MSc degrees from the Nanjing University of Science Technology, China, and the PhD degree from Southeast University. He is currently a full professor with the Center for Research on Intelligent System and Engineering, Institute of Automation, Chinese Academy of Sciences. He is also with the University of Chinese Academy of Sciences (UCAS), and the CAS Center for Excellence in Brain Science and Intelligence Technology. He has authored or coauthored more than 210 papers in the important international journals and conferences, such as the IEEE TPAMI, T-IP, T-SMCB, TCSVT, Pattern Recognition, CVIU, ICCV, ECCV, CVPR, ICIP, and ICPR. His current research interests include computer vision, pattern recognition and game theory, including object recognition, video analysis, and visual surveillance. He is the co-chair and program committee member more than 40 international conferences, such as ICCV, CVPR, ECCV, and the IEEE workshops on visual surveillance. He is an associate editor for *IEEE Transactions on Systems, Man, and Cybernetics: Systems* and *Pattern Recognition*.