International Journal of Computer Applications in Technology

ISSN online: 1741-5047 - ISSN print: 0952-8091
https://www.inderscience.com/ijcat

Mean-shift-based moving target tracking algorithm in complex industrial environments

Zhongming Liao, Zhaosheng Xu, Xiuhong Xu, Azlan Ismail

DOI: 10.1504/IJCAT.2025.10074104

Article History:

Received:	26 February 2025
Last revised:	21 July 2025
Accepted:	18 September 2025
Published online:	17 February 2026

Mean-shift-based moving target tracking algorithm in complex industrial environments

Zhongming Liao and Zhaosheng Xu

School of Mathematics and Computer Science,
Xinyu College,
Xinyu, Jiangxi, China
and
College of Computing, Informatics and Mathematics,
Universiti Teknologi MARA (UiTM),
Shah Alam, Selangor, Malaysia
Email: liaozhongming@xyc.edu.cn
Email: xyxyxzs2018@163.com

Xiuhong Xu*

School of Vocational and Continuing Education,
Xinyu College,
Xinyu, Jiangxi, China
Email: xxh258639049@163.com
*Corresponding author

Azlan Ismail

College of Computing, Informatics and Mathematics,
Universiti Teknologi MARA (UiTM),
Shah Alam, Selangor, Malaysia
and
Institute for Big Data Analytics and Artificial Intelligence (IBDAAI),
Kompleks Al-Khawarizmi,
Universiti Teknologi (UiTM),
Shah Alam, Selangor, Malaysia
Email: azlanismail@uitm.edu.my

Abstract: This paper proposes an improved moving Target Tracking Algorithm (TTA) based on the Mean-Shift (MS) method, which is suitable for complex industrial environments. The improved algorithm introduces the You Only Look Once (YOLO) model for moving target detection and uses its results as tracking input. In addition, the algorithm also introduces a twin network (SN) to extract the deep features of the target for re-identification after occlusion. In order to further improve the tracking stability, a Kalman Filter is introduced to predict the next motion state of the target. Stability analysis shows that the algorithm achieves the best Multi-target Tracking Accuracy (MOTA) index in various complex environments, outperforming other tracking methods and showing good multi-target tracking stability. In summary, the algorithm successfully overcomes the limitations of the traditional MS method and provides a novel solution for moving target tracking in industrial environments. The algorithm has important practical value and provides a valuable reference for future research on moving target tracking in dynamic and complex environments.

Keywords: moving target tracking; mean-shift algorithm; YOLO model; Siamese network; Kalman Filter.

Reference to this paper should be made as follows: Liao, Z., Xu, Z., Xu, X. and Ismail, A. (2026) 'Mean-shift-based moving target tracking algorithm in complex industrial environments', *Int. J. Computer Applications in Technology*, Vol. 78, No. 2, pp.112–123.

Biographical notes: Zhongming Liao is a Doctor and Graduated from East China Normal University majoring in Computer Science and Technology. Now studying in Mara University of technology, he works in Jiangxi Xinyu College. His research interests include computational intelligence, information security, platform construction, machine learning, deep learning, big data analysis and computer algorithm research.

Zhaosheng Xu obtained a Master's degree in Computational Mathematics from Anhui University in 2010 and Bachelor degree of Mathematics and Applied Mathematics in 2007. Starting to study for a Doctorate in 2021. In 2010, he worked in Xinyu College, engaged in teaching and research management and teaching work. His research interests include algorithm design and higher education reform and data analysis.

Xiuhong Xu graduated in Computer Science and Technology from Jiangxi Normal University in China and is currently working at Xinyu College. Her main research interests include artificial intelligence, information security, platform construction, image processing and big data analysis.

Azlan Ismail received his BSc degree and MSc degree in Computer Science from Universiti Teknologi Malaysia. Then, he received PhD degree from the University of Wollongong, Australia in 2012. Currently, he is a senior fellow of the Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Universiti Teknologi MARA (UiTM). He is also an Associate Professor in the Department of Computer Science, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM). He had conducted his Postdoctoral study in the Department of Computer Science at the University of Oxford which was funded by the Ministry of Higher Education, Malaysia. His research interests include cloud computing, machine learning, self-adaptive systems and software verification. He has served as a Reviewer for international journals/conference proceedings and has also been involved in the organising committee for national and international conferences.

1 Introduction

Recent advancements in industrial automation and the Industrial Internet of Things (IIoT) have increased the demand for real-time, highly accurate target tracking in complex industrial environments. These settings, often featuring dynamic and noisy backgrounds, pose challenges for traditional computer vision algorithms. Moving target tracking, particularly in such environments, has thus become a critical research focus (Zou et al., 2022; Ram, 2022).

Industries like manufacturing, robotics and surveillance rely heavily on computer vision systems to monitor and track moving targets, such as machines, vehicles and workers. However, factors like illumination variations, occlusions, cluttered backgrounds and rapid changes in target scale and orientation hinder system performance. Illumination fluctuations in factory or outdoor settings impact the robustness of traditional methods, while occlusions – due to physical obstructions or overlapping targets – complicate continuous tracking. Non-uniform motion patterns further complicate tracking algorithms. Classical techniques, such as the Kalman Filter or Mean-Shift (MS) algorithm, often struggle to maintain accuracy under these conditions (Maghraby et al., 2023).

In response, research has shifted toward deep learning-based methods to better handle these challenges. Models like DeepSORT and MS variants with noise filtering (MS+SN) or Kalman Filters have shown promise, but they still struggle with scale variance, occlusion and real-time performance. Combining classical tracking methods with modern deep learning techniques offers a promising solution, merging the strengths of both in terms of accuracy, robustness and speed.

This paper presents a hybrid approach that combines deep learning models with classical tracking algorithms to address the unique challenges posed by complex industrial environments. Our model integrates deep learning for feature extraction and tracking with established methods like MS and Kalman Filtering for prediction and state updates. The main objectives are to enhance tracking accuracy, improve re-identification after occlusions and increase system robustness. Experimental results demonstrate that the proposed algorithm significantly outperforms existing methods, including the classical MS algorithm, DeepSORT and MS+SN and MS+Kalman models. It achieves superior tracking precision, recall and F1-scores in various test scenarios. Notably, its ability to maintain accurate tracking during occlusions highlights its potential for real-world industrial applications.

2 Related work

Many scholars have proposed different algorithms and methods for the problem of moving target tracking. Ma et al. (2022) developed a distributed extended state observer that incorporates a switching topology for the cooperative tracking of targets by autonomous surface vehicles. The observer is used to comprehensively estimate the dynamics of unknown targets and the dynamics of adjacent autonomous surface vehicles. Wang and Liu (2022) proposed a method to improve monitoring accuracy based on real-time operation of YOLOv3 to solve the problem of complex background and mutual obstruction between multiple targets in the process of autonomous driving, and achieved real-time warning for completely obstructed objects. Liu et al. (2023) introduced fuzzy reasoning into the tracking process for the remote monitoring problem in intelligent transportation systems, analysed the reliability of detection images and experimental results on multiple data sets showed that the proposed algorithm outperformed other similar algorithms in multiple evaluation indicators. However, although these methods have certain advantages in theory, they often face problems such as target occlusion, background interference and illumination changes in actual industrial applications, and still fail to provide sufficient robustness and accuracy.

Although the existing tracking algorithms based on MS can cope with target tracking tasks to a certain extent, Han et al. (2022) proposed a TTA based on adaptive bandwidth MS, which generates anisotropic kernel functions by introducing signed distance constraint functions. Based on the MS window centre calculation method and the similarity threshold adaptive template update, experiments show that the proposed algorithm has superiority. Given the low accuracy and long-time consumption of traditional methods for tracking aerobics athletes' facial images, Yang (2022) proposed a facial feature tracking algorithm based on Kalman filtering and MS. The experimental results demonstrate that the proposed method achieves a tracking accuracy of 97%, with the shortest tracking time approximately 1.5 s. Their limitations in complex industrial environments are still prominent. First, the traditional MS algorithm has poor adaptability to lighting changes, background interference and target occlusion. Second, the scale change and target re-identification problems have not been effectively solved. In addition, traditional methods usually fall into local optimal solutions, and it is difficult to maintain stable tracking effects in complex environments. Given the limitations of the traditional MS algorithm (Rani et al., 2022; Kumah et al., 2022; Yang et al., 2023), this paper improves the MS algorithm. The YOLO model (Terven et al., 2023; Diwan et al., 2023; Lee and Hwang, 2022) is used to detect moving targets in the initialisation of the MS algorithm, and the detected target box is used as the initialisation input of the improved algorithm. The SN (Li et al., 2022; Javed et al., 2022; Wu et al., 2020) extracts deep features of the moving target by combining with the target box detected by the YOLO model. At the same time, the SN can also help improve the algorithm to improve the ability to re-identify moving targets after they are occluded. During tracking, the algorithm employs Kalman filtering (Khodarahmi and Maihami, 2023; Chen et al., 2022; Gurin et al., 2024) to forecast the future state of a moving target, which enhances its ability to track the target with greater accuracy and stability.

3 Moving target tracking and optimisation method based on complex industrial environment

3.1 Introduction to MS algorithm

The MS algorithm is a density-based non-parametric clustering algorithm, which is widely used in target tracking, image segmentation and other tasks. The core concept of the algorithm is as follows: for each data point, move along the density gradient direction of the data until the local maximum point of the data is found. In the moving target tracking of this paper, the data point represents the pixel of the moving target, and the density represents the degree of aggregation of the target features around the pixel.

The MS algorithm uses kernel density estimation to calculate the density, and the calculation formula is:

$$\hat{f}(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

In this context, n refers to the number of data points, h is the bandwidth parameter that controls the domain size and K represents the kernel function. This study utilises the Gaussian kernel function. The calculation of the Gaussian kernel function is expressed as:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} \tag{2}$$

In the moving target tracking of this paper, the density function represents the feature histogram of the moving target.

For each point x in the target area, the weighted average position of the point in the current window must be calculated. This position is called the MS vector, and the calculation formula is:

$$m(x) = \frac{\sum_{i=1}^{n} x_i K\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)} \tag{3}$$

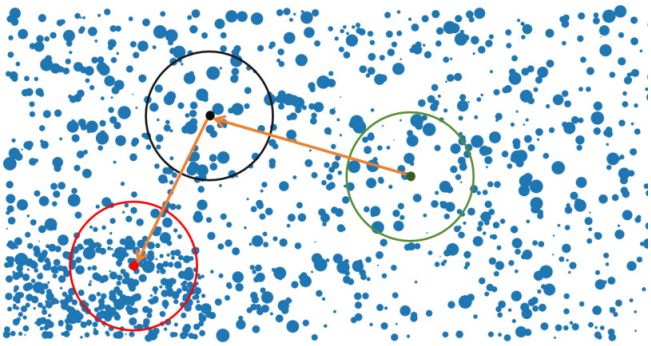
x represents the current point, x_i represents the sample point and $K\left(\frac{x - x_i}{h}\right)$ represents the weighting factor of the kernel function, which controls the influence of the sample point on the target position update. Then, the updated position is iterated through the following update rule:

$$x_{k+1} = x_k + m(x_k) \tag{4}$$

x_k represents the point of the k-th iteration, and $m(x_k)$ represents the MS vector calculated based on the current point. The algorithm continues to iterate until the termination condition is reached.

The MS algorithm iteration is shown in Figure 1.

Figure 1 MS algorithm iteration diagram (see online version for colours)



3.2 YOLO design

YOLO is a deep learning-based target detection method renowned for its efficient real-time performance and robust accuracy. Unlike traditional approaches, which break down the detection task into multiple sub-problems, YOLO treats it as a regression problem. It employs a single neural network to simultaneously predict both the location and category of the target. This end-to-end training method gives YOLO a significant advantage in processing speed.

In complex industrial environments, accurately identifying and locating target objects is critical, especially when multiple targets need to be detected simultaneously. YOLO's efficient real-time detection capability excels in handling challenging industrial scenarios, such as target occlusion, background interference and varying lighting conditions.

Assuming that the input target image is $I = R^{W\times H\times 3}$, first, resize the input image to a fixed size for easier processing by the YOLO model. Then, normalise the pixel values to the range of [0, 1] using the normalisation formula:

$$I_{norm}(x,y) = \frac{I(x,y)}{255} \tag{5}$$

$I(x,y)$ represents the pixel value of the input image at position (x,y), and $I_{norm}(x,y)$ represents the normalised pixel value. This normalisation operation is applied to each pixel point individually.

After that, it is divided into grids. Each grid predicts the likelihood of the target's presence and its position within the grid. The size of the divided grid is:

$$GS = \frac{W}{S}\times\frac{H}{S} \tag{6}$$

W and *H* denote the width and height of the input image, with *S* representing the number of grids.

Each grid predicts multiple bounding boxes, where the output of each bounding box includes:

1) *The target centre's coordinates*: A description of its position on the grid.
2) *Width and height*: The width and height of the bounding box represent its normalised size relative to the image.
3) *Confidence*: The probability of whether the grid prediction contains the target object.
4) *Category probability distribution*: The probability of predicting the target category.

Therefore, the output of each grid can be expressed as:

$$\left(x, y, w, h, P_{object}, c_1, c_2, ..., c_k\right) \tag{7}$$

The combination of *x* and *y* represents the location of the object, and the combination of *w* and *h* represents the width and height of the bounding box, P_{object} represents the probability that the grid contains the target, and $c_1, c_2, ..., c_k$ represents the category probabilities.

YOLO optimises the network by reducing the difference between predicted and true labels. The loss functions include the following three categories.

Position loss indicates the difference between the detected target position and the true position:

$$\begin{aligned} L_C = \lambda_c \sum_i l_0 \Big(&(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\ &+ (w_i - \hat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2 \Big) \end{aligned} \tag{8}$$

Among them λ_c is a hyperparameter in the position loss function.

Confidence loss: This loss is used to measure the confidence of the prediction and the actual existence of the target:

$$L_o = \lambda_o \sum_i l_o \left(P_i - \hat{P}_i\right)^2 + \lambda_n \sum_i l_n \left(P_i - \hat{P}_i\right)^2 \tag{9}$$

Among them λ_o is a hyperparameter in the confidence loss function.

Category loss: This loss calculates the difference between predicted and actual categories, using cross-entropy loss for computation:

$$L_{class} = \lambda_{class} \sum_i l_o \sum_{c=1}^{C} (c_i - \hat{c}_i)^2 \tag{10}$$

Among them λ_{class} is a hyperparameter in the category of loss function.

Combining the above three types of losses, the final total loss function can be expressed as:

$$L_{total} = L_C + L_o + L_{class} \tag{11}$$

In the output of YOLO in this paper, there may be multiple overlapping bounding boxes. To address this situation, this paper adopts the Non-Maximum Suppression Strategy (NMS) (Oro et al., 2022) to remove duplicate detection results. The implementation steps of NMS are:

1) Calculate the confidence of each candidate's bounding box.
2) Sort the confidences and select the bounding box with the highest confidence as the current detection result.
3) Eliminate bounding boxes with an intersection-over-union ratio higher than the threshold of the current bounding box, keeping the one with the highest confidence.
4) Repeat the above steps until all bounding boxes are processed.

The detection module incorporates multi-scale feature fusion from different network layers to handle target scale variations. Shallow layer features preserve finer spatial details for small targets, while deep layer features capture semantic information for larger targets. The anchor boxes are dynamically adjusted based on the statistical distribution of

target sizes in the training data set. During tracking, the mean-shift kernel bandwidth is adaptively scaled according to the detected target dimensions from YOLO, ensuring consistent performance across scale transformations. This integrated approach maintains detection and tracking robustness without requiring explicit scale estimation steps.

3.3 SN design

SN is a deep learning model for target re-identification tasks. The main task of this model is to achieve identity recognition under different perspectives, lighting or backgrounds. The specific method is to judge whether they belong to the same target by comparing the similarity of the targets. In industrial environments, targets often experience occlusion, appearance changes and perspective changes. SNs can effectively handle these problems. Through deep feature learning, SNs can effectively restore occluded targets and accurately re-identify targets even when their appearance changes.

The SN consists of two identical sub-networks. This paper uses a deep Convolutional Neural Network (CNN) (Cong and Zhou, 2023; Bharadiya, 2023) as a sub-network. The two sub-networks share weights and receive two input images. The SN calculates the similarity of the two input images and outputs whether they represent the same target. For each input image *x*, after passing through the shared deep convolutional neural network, the output features are:

$$f(x) = CNN(x) \tag{12}$$

CNN stands for a deep convolutional neural network.

The two feature vectors $f(x_1)$ $f(x_2)$, after feature extraction, are then sent to the Siamese similarity measurement layer, where the distance between the two feature vectors is calculated using the Euclidean distance formula:

$$D_w = \| f(x_1) - f(x_2) \|_2 = \sqrt{\sum_{i=1}^{d} \left(f_i(x_1) - f_i(x_2) \right)^2} \tag{13}$$

Among them, $f_i(x_1)$ $f_i(x_2)$ they represent the *i*-th element in vectors $f(x_1)$ and $f(x_2)$ respectively. The smaller the Euclidean distance D_w is, the more similar the two feature vectors are.

Based on the calculated similarity metric, Siamese output generates a similarity score to determine if the two input images belong to the same target. The specific output layer is a Sigmoid activation function, which is used to output the similarity score:

$$\hat{y} = \sigma(D_w) \tag{14}$$

$\sigma(x)$ It is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

The similarity score output by the output layer is between 0 and 1, indicating the probability that the two input images are the same object. If the similarity score output is close to 1, it means that the two images are similar, indicating that the two input images belong to the same object. If it is close to 0, it means that the two input images do not belong to the same object.

The goal of the SN is to minimise the distance for the same target while maximising the distance for different targets. Contrast loss serves as the loss function during network training. The calculation formula is:

$$L(y, \hat{y}) = \frac{1}{2}\left[y \cdot D_w^2 + (1 - y) \cdot \max(0, m - D_w)^2 \right] \tag{16}$$

During training, y serves as the label, determining whether two images are the same object (the label value is 1 for the same target, and 0 for not), D_w represents the Euclidean distance between the features of the two input images, and m represents the threshold, which is used to set the minimum distance between different targets. $y \cdot D_w^2$ In contrast, the loss function is used to minimise the distance to the same target, and $(1 - y) \cdot \max(0, m - D_w)^2$ is used to maximise the distance between different targets.

The current framework employs temporal consistency verification to handle partial occlusion scenarios. When occlusion is detected through feature dissimilarity, the algorithm maintains the target's motion trajectory using Kalman prediction while suspending template updates. For severe occlusion cases, the system activates a short-term memory buffer that stores pre-occlusion appearance features. This buffer enables more reliable re-identification once the target reappears by comparing against historical feature representations rather than immediately updating with potentially corrupted occlusion-period features.

The Siamese network training process implemented several optimisation strategies to improve feature discriminability. Data augmentation methods included random cropping with scale variations between 0.8 and 1.2 and colour jittering with brightness adjustments up to 30% of the original values. The contrastive loss function incorporated an adaptive margin that increased from 0.5 to 1.2 based on training epoch progression. Network architecture modifications employed residual skip connections in the convolutional layers to facilitate gradient propagation. Training batches were constructed using a hard example mining strategy that maintained a 1:3 ratio of positive to negative pairs. Learning rate scheduling followed a cosine annealing pattern between 1e-4 and 1e-3 across training epochs. During training, the parameters of the SN are adjusted to reduce contrast loss and enhance the feature extraction capability of the target.

During training, the parameters of the SN are adjusted to reduce contrast loss and enhance the feature extraction capability of the target. Specific parameter adjustments are presented in Table 1.

Table 1 SN parameter adjustment

Parameter	*Description*	*Value*
Number of convolutional layer filters	The number of convolution kernels (filters) in each layer of a convolutional neural network	64
Convolution kernel size	The size of the convolution kernel used in the convolution operation	3x3
Pooling layer size	The size of the pooling operation	2×2
Fully connected layer dimensions	The output dimension of the last fully connected layer	256
Minimum distance threshold	The threshold used in the contrastive loss	1.0
Learning rate	Learning rate of the optimiser	0.001
Batch size	The number of samples for each training	32
Number of training rounds	The total number of rounds of network training	100

The current parameter settings were determined through empirical analysis. Future implementations could employ automated hyperparameter optimisation techniques to systematically explore the parameter space. Bayesian optimisation and random search methods offer principled approaches to identify optimal configurations without relying on manual tuning, potentially improving model convergence and tracking performance. The current Siamese network architecture can be extended to incorporate feature weighting based on the attention mechanism in future implementations. The extension will allow the network to dynamically emphasise target-specific regions during feature extraction while suppressing background interference. The attention mechanism will work in conjunction with the existing convolutional layers to add adaptive spatial weighting while preserving the original feature extraction process. This enhancement will improve feature discriminability in cluttered environments without requiring fundamental changes to the network structure or training protocol.

3.4 Kalman Filter combined with MS

The Kalman Filter is a recursive linear filtering technique used for estimating linear states. It works by providing the optimal state estimate, minimising the variance of the estimation error. The Kalman Filter mainly includes two steps: prediction and update.

Prediction is divided into predicted state and predicted covariance. The formula for the predicted state is:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{17}$$

Among them, $\hat{x}_k^-$ A represents the state transfer matrix, $A\hat{x}_{k-1}$ B represents the control input matrix and u_k represents the control input.

The formula for the predicted covariance is:

$$P_k^- = AP_{k-1}A^T + Q \tag{18}$$

P_k^- represents the prediction error covariance, and Q represents the process noise covariance matrix.

The update step involves computing the Kalman gain, adjusting the state estimate and modifying the error covariance. The formula for calculating the Kalman gain is:

$$K_k = P_k^- H^T \left(HP_k^- H^T + R \right)^{-1} \tag{19}$$

K_k represents the Kalman gain, H is the observation matrix and R is the observation noise covariance matrix.

The formula for updating the state estimate is:

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H\hat{x}_k^- \right) \tag{20}$$

$\hat{x}_k$ is the updated state estimate, and z_k is the observation at the current moment.

The formula for updating the error covariance is:

$$P_k = \left(I - K_k H \right) P_k^- \tag{21}$$

P_k is the updated error covariance.

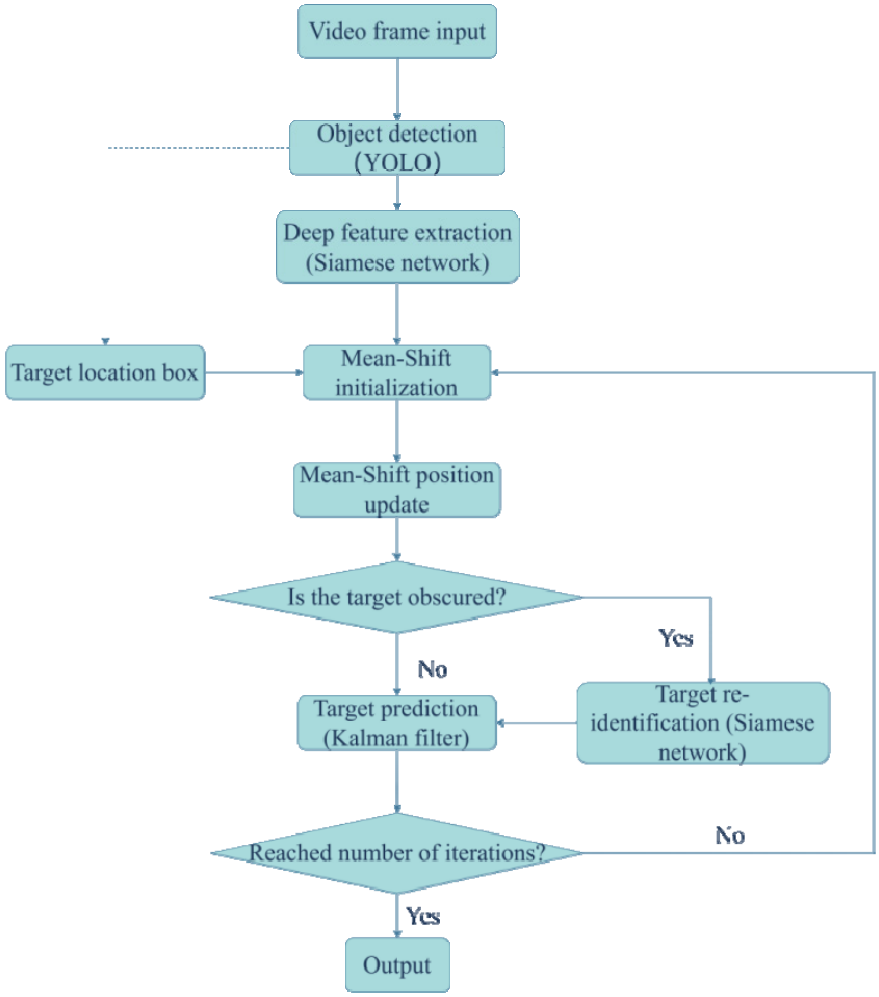
The paper combines Kalman filtering with MS, allowing both algorithms to complement each other, addressing the limitations of each in complex industrial environments. In this study, the Kalman Filter algorithm predicts and smooths the target state, especially during short-term occlusion or rapid motion. It forecasts the target's next position and corrects tracking errors when the target loses its original point. MS accurately detects the target's position, locating it in each frame of motion using appearance information.

The Kalman Filter incorporates a dynamic adjustment mechanism for its prediction and update parameters based on the target's motion state and environmental conditions. The process noise covariance matrix Q and observation noise covariance matrix R are adaptively updated according to the target's motion consistency and feature matching confidence. When the target exhibits rapid or nonlinear motion, the system increases Q to account for higher uncertainty in state prediction. Conversely, stable motion reduces Q to enhance prediction precision. Similarly, R is adjusted based on the reliability of visual observations, decreasing when the Siamese network yields high similarity scores and increasing under occlusion or low-confidence detections. This adaptive mechanism ensures robust tracking performance across varying motion patterns and environmental interference.

3.5 Comprehensive optimisation of MS

The optimised MS optimisation framework based on moving targets in complex industrial environments is shown in Figure 2.

Figure 2 Optimisation algorithm flow chart (see online version for colours)



To address the problem of target scale variation, the algorithm integrates multi-scale feature fusion in target representation. YOLO provides an initial scale estimate, while the mean shift kernel bandwidth is dynamically adjusted based on the spatial distribution of the target. The twin network extracts features from multiple receptive fields, ensuring scale-invariant tracking without explicit scale estimation.

YOLO provides high-confidence detection regions as input to the twin network. When the prediction uncertainty of the Kalman Filter exceeds a set threshold, the system selectively activates feature extraction. The motion update of the Kalman Filter is weighted by the similarity score of the twin network, forming a feedback loop that dynamically adjusts the reliability of the prediction based on the confidence of the visual match. This conditional strategy optimises computational efficiency while maintaining tracking accuracy. In addition, the architecture supports multi-stage parallel processing. YOLO's detection is batch-processable, the Siamese network's convolutional features can be distributed across GPUs and Kalman Filter predictions are inherently parallel in multi-target scenarios. This design paves the way for distributed implementations, where modules can operate as micro-services, communicating through a message queue for scalable processing.

The algorithm's parameter sensitivity was evaluated through systematic testing of key components. The mean-shift kernel bandwidth demonstrated stable tracking performance within 0.5–1.5 pixel range, with optimal MOTA scores achieved at 0.8 bandwidth setting. Kalman Filter parameters showed robustness when process noise covariance values were maintained between 0.1 and 0.3 and measurement noise between 1.0 and 3.0. The Siamese network similarity threshold exhibited consistent re-identification accuracy when configured between 0.7 and 0.85 confidence level. These parameter ranges were determined through exhaustive testing across all experimental scenarios, with performance variations remaining below 5% within the specified boundaries.

4 Experiment design

4.1 Experiment data

The experiment is based on the VIRAT data set, which is a video data set focusing on complex environments (industrial workshops, warehouses, parking lots, etc.). Some of the experimental data are shown in Figure 3, and the description of some of the data sets is shown in Table 2.

Figure 3 Experiment environment (see online version for colours)

Table 2 Description of some data set contents

Video clips	*Target type*	*Video content*
1	Pedestrians, forklifts, robotic arms	Industrial workshop scene
2	Pedestrians and objects	Warehouse scene
3	Vehicles, pedestrians	Production line entrance
4	Pedestrians, objects, machinery and equipment	Industrial workshop
5	Vehicles, pedestrians, objects	storehouse
6	Pedestrians, robotic arms and tools	Automated production line
7	Vehicles, pedestrians, robots	Industrial workshop scene
8	Pedestrians and objects	Warehouse scene
9	Pedestrians, forklifts, conveyor belts	Production line entrance

The video data set contains videos of different industrial scenes, such as workshops, warehouses and production lines. The target labels for tracking are different in different videos.

The experiment was further expanded to include more industrial scene videos, covering a variety of complex scenarios such as extreme lighting changes, fast target motion and high-density target overlap. The purpose of selecting these scenes is to rigorously test the robustness of the algorithm under various challenging conditions. Extreme lighting changes include low-light and strong glare environments, fast motion scenes simulate situations where the target moves much faster than a typical industrial environment, and high-density target overlap scenes present a typical crowded industrial environment where targets often occlude each other. By introducing these diverse conditions, the aim is to comprehensively evaluate the performance and adaptability of the algorithm in a wider range of industrial applications.

The experiment evaluation included multi-target interaction scenarios to assess tracking performance under complex conditions. The test cases incorporated frequent occlusions and potential collisions between targets, with tracking accuracy measured during these interaction events. The data set contained sequences with varying target densities to evaluate the algorithm's ability to maintain distinct identities during prolonged occlusions. Motion patterns were designed to induce trajectory crossings and spatial proximity between targets, challenging the re-identification capability of the tracking system. These scenarios provided quantitative measures of identity preservation and tracking continuity in multi-target environments.

4.2 Comparative experiment

The algorithm implemented in this paper is compared with the following algorithm:

1) Traditional MS algorithm
2) DeepSORT algorithm: This algorithm combines the target appearance information and motion information and can effectively track multiple targets.
3) MS+SN
4) MS+Kalman Filter

4.3 Evaluation indicators

The accuracy measures the degree of overlap between the predicted target position and the actual marked position:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

Recall measures how many of all true targets are correctly tracked:

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

The F1-score is used to comprehensively evaluate the precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{24}$$

The accuracy of multi-target tracking is evaluated by Multiple Object Tracking Accuracy (MOTA). In multi-target tracking, factors such as target loss, misidentification and target mismatch are considered. The evaluation formula is:

$$\text{MOTA} = 1 - \frac{\sum(\text{Misser} + \text{Fps} + \text{Switches})}{\text{Total Ground Truth Objects}} \tag{25}$$

Misser represents the frequency of target loss, Fps indicates the instances of target misidentification and Switches refers to the occurrences of target identity swapping.

The accuracy of target identity preservation is measured by the IDF1 indicator:

$$IDF1 = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{26}$$

The experiment further evaluated the algorithm's cross-scenario adaptability by dynamically switching between different industrial environments during tracking. The test involved transitioning from a warehouse scene to a production line entrance, followed by an industrial workshop setting, without resetting the tracking parameters. This design assessed the algorithm's ability to maintain tracking

performance when environmental conditions changed abruptly. The evaluation metrics remained consistent with the original experiment to ensure comparability of results. The dynamic scenario transitions tested the robustness of the feature extraction and motion prediction components under varying background conditions and target distributions.

5 Result analysis

5.1 Tracking accuracy analysis

In industrial environments, high-precision tracking algorithms are essential to improving production efficiency and safety. To test the effectiveness and robustness of the improved algorithm in complex environments, it is ensured that it can still accurately track the target under the influence of factors such as lighting changes, occlusion and background interference, the tracking accuracy of the improved algorithm is compared with the traditional MS algorithm, DeepSORT algorithm, MS+SN and MS+Kalman Filter algorithm. Results are presented in Table 3.

Table 3 Tracking accuracy

Algorithm	*Precision*	*Recall*	*F1-Score*
MS algorithm	0.76	0.74	0.75
DeepSORT algorithm	0.82	0.84	0.83
MS + SN	0.87	0.83	0.85
MS + Kalman Filter	0.84	0.78	0.81
The improved algorithm in this paper	0.92	0.88	0.90

As can be seen from Table 3, the improved algorithm implemented in this paper performs well in tracking accuracy. The precision and recall of the improved algorithm are significantly higher than those of other algorithms, indicating that the algorithm can accurately track targets in complex environments, occlusions and appearance changes, while also effectively capturing most targets. At the same time, the F1-score of the improved algorithm is also the highest, which means that the improved algorithm has the best balance between precision and recall, and can meet the high-demand moving target tracking tasks in industrial environments. Compared with the improved algorithm, the traditional MS algorithm and MS+Kalman Filter algorithm may have problems of false tracking and missed tracking in actual applications due to their low precision and recall. Although the DeepSORT algorithm and MS+SN have improved performance compared to the first two algorithms, they are still not as good as the improved algorithm in this paper. In summary, the improved algorithm in this paper has obvious advantages and is suitable for moving target tracking tasks in complex scenes in industrial environments.

5.2 Occlusion and re-identification analysis

In the moving target tracking task studied in this paper, occlusion and re-identification capabilities are crucial. Especially in complex industrial environments, occlusion can lead to target loss or mistracking, while re-identification capabilities determine the recovery and accurate tracking of target identities after occlusion. The analysis of occlusion and re-identification capabilities helps optimise the algorithm's extraction of target features and identity recovery measurements, improves tracking stability and accuracy and reduces the problems of mistracking and target loss. The occlusion and re-identification capabilities of the algorithm can be analysed by comparing the IDF1 index, and the results are shown in Figure 4.

As shown in Figure 4, the IDF1 values for all algorithms dropped after occlusion, including the improved one proposed in this paper, indicating that occlusion weakened the ability to maintain the target identity. The improved algorithm has the smallest change in IDF1 value after occlusion, showing a strong target re-identification ability. Compared with the improved algorithm, the other four algorithms have larger changes in IDF1 value after occlusion. The IDF1 value changes of the DeepSORT algorithm, MS+SN and MS+Kalman Filter algorithm after occlusion are similar, with change values of 0.07, 0.07 and 0.06, respectively. The MS algorithm has the largest change in IDF1 value after occlusion among the four algorithms, with a change value of 0.1. The combined algorithm that introduces the SN and the Kalman algorithm on this basis has a smaller change in IDF1 value after occlusion than the MS algorithm. This shows that the introduction of the SN and the Kalman algorithm can effectively improve the algorithm's re-identification ability. This shows the limitations of the single MS algorithm.

5.3 Algorithm stability and robustness analysis

Since complex industrial environments usually have factors such as illumination, target occlusion, appearance changes and background interference, it is necessary to perform stability and robustness analysis on the improved algorithm. The specific analysis method is to design different scenario variables to analyse the adaptability of different algorithms under complex environmental conditions and their consistency in long-term tracking, and use the MOTA indicator to measure the results. The analysis results are shown in Figure 5.

Figure 4 Occlusion and re-identification results (see online version for colours)

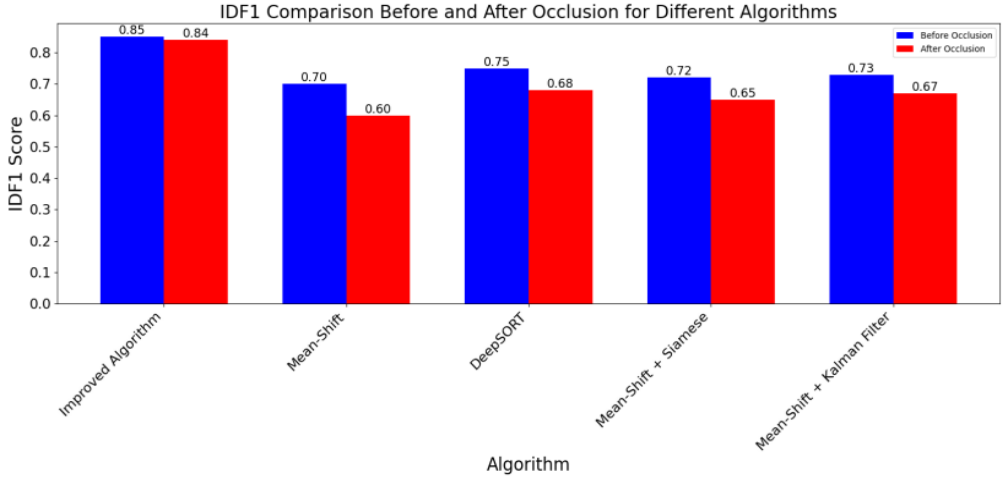
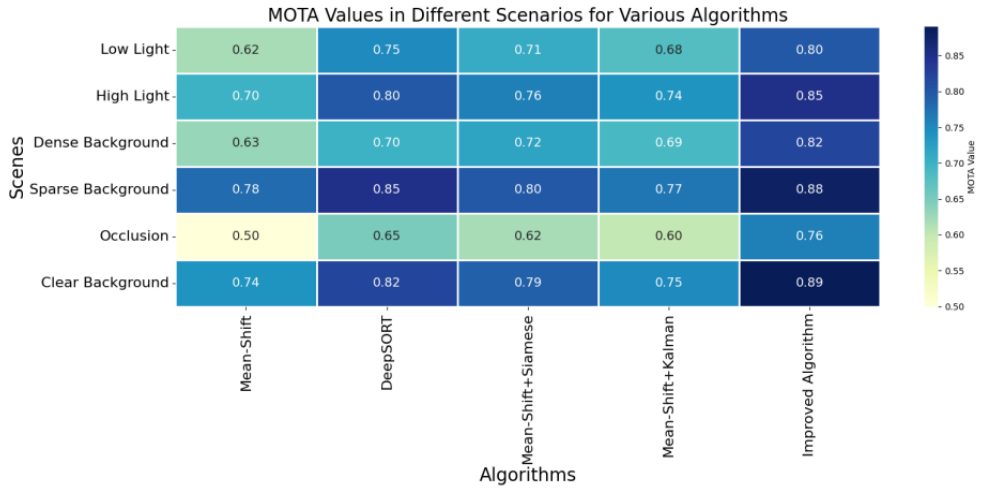


Figure 5 Stability and robustness analysis results (see online version for colours)



Based on the experimental results in Figure 5, performance varies considerably across different algorithms and scenarios. In the clear background, the MOTA values of each algorithm are relatively close, and the MOTA value of the improved algorithm is the highest, which is 0.89. In other complex scenarios, such as low light, high light, sparse background, dense background and occlusion, the MOTA value of the improved algorithm is still the highest. The MOTA values are 0.80, 0.85, 0.88, 0.82 and 0.76, respectively, indicating that the improved method in this paper, combined with YOLO target detection, Siamese re-identification network and Kalman Filter MS improved algorithm, can effectively improve the stability and robustness of multi-target tracking. This also shows that the improved algorithm can provide stable and accurate multi-target tracking functions in complex industrial environments.

6 Conclusions

This paper designs an improved algorithm based on the MS algorithm. The specific improvements are to introduce YOLO for target detection on the basis of the MS algorithm, initialise the algorithm input according to the detected target frame, introduce Siamese for deep feature extraction of moving targets and improve the algorithm's ability to re-identify after occlusion. Kalman filtering can be introduced to predict moving targets and assist the algorithm in tracking moving targets, thereby improving the stability of the improved algorithm. The experimental results show that the improved algorithm in this paper has excellent tracking accuracy of moving targets in complex environments, the ability to re-identify targets after occlusion and the stability and robustness of the algorithm. Although the improved algorithm in this paper provides a new idea for the problem of tracking moving targets in complex industrial environments, this study still needs to be improved in some aspects:

1) The improved algorithm in this paper combines YOLO for moving target detection, SN for moving target re-identification and an experimental Kalman Filter for target prediction. However, the deep learning model used has a large amount of calculation, and there are certain difficulties in calculation efficiency in actual tracking scenarios, which may make it difficult to meet real-time requirements.

2) The deep learning model used in this paper depends largely on the quality of the data set and the hyperparameter settings during the training process. In complex, specific industrial environments, it may face the problem of limited training data and weak algorithm generalisation ability.

3) This paper combines multiple deep learning models and traditional methods, which can enhance the robustness and accuracy of the algorithm, but also adds to its complexity. At the same time, the collaborative work between different models may lead to additional computing resource consumption, and the tuning of parameters between models may also become complicated.

The future can focus on exploring model compression techniques such as quantisation and pruning to reduce computational overhead while maintaining tracking accuracy. Additionally, research can investigate the potential for incorporating automatic hyperparameter optimisation to automatically identify the optimal hyperparameter combination, thereby improving model training performance. The computational efficiency of the proposed algorithm presents an area for future improvement. Research directions could explore the development of lightweight network architectures through techniques like depthwise separable convolutions or channel pruning. The reduction of model parameters while maintaining tracking accuracy would enhance real-time performance in industrial applications. Network quantisation methods may be investigated to decrease memory usage and computational load without significant degradation in tracking precision.

Acknowledgement

This work is supported in part by the XScience and Technology Research Project of Jiangxi Provincial Department of Education [grant number GJJ2202215]. We thank all the anonymous reviewers who generously contributed their time and efforts. Their professional recommendations have greatly enhanced the quality of the manuscript.

Declarations

All authors declare that they have no conflicts of interest.

References

Bharadiya, J. (2023) 'Convolutional neural networks for image classification', *International Journal of Innovative Science and Research Technology*, Vol. 8, No. 5, pp.673–677.

Chen, Y., Sanz-Alonso, D. and Willett, R. (2022) 'Autodifferentiable ensemble Kalman Filters', *SIAM Journal on Mathematics of Data Science*, Vol. 4, No. 2, pp.801–833.

Cong, S. and Zhou, Y. (2023) 'A review of convolutional neural network architectures and their optimizations', *Artificial Intelligence Review*, Vol. 56, No. 3, pp.1905–1969.

Diwan, T., Anirudh, G. and Tembhurne, J.V. (2023) 'Object detection using YOLO: challenges, architectural successors, datasets, and applications', *Multimedia Tools and Applications*, Vol. 82, No. 6, pp.9243–9275.

Gurin, D., Yevsieiev, V., Abu-Jassar, A. and Maksymova, S. (2024) 'Using the Kalman Filter to represent probabilistic models for determining the location of a person in collaborative robot working area', *Multidisciplinary Journal of Science and Technology*, Vol. 4, No. 8, pp.66–75.

Han, M., Wang, J., Wang, J., Meng, J. and Cheng, Y. (2022) 'Research on a target tracking algorithm based on mean shift with adaptive bandwidth', *Journal of Computational Methods in Sciences and Engineering*, Vol. 22, No. 2, pp.661–675.

Javed, S., Danelljan, M., Khan, F.S., Khan, M.H., Felsberg, M. and Matas, J. (2022) 'Visual object tracking with discriminative filters and Siamese networks: a survey and outlook', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, No. 5, pp.6552–6574.

Khodarahmi, M. and Maihami, V. (2023) 'A review on Kalman Filter models', *Archives of Computational Methods in Engineering*, Vol. 30, No. 1, pp.727–747.

Kumah, C., Zhang, N., Raji, R.K., Li, Z. and Pan, R. (2022) 'Unsupervised segmentation of printed fabric patterns based on mean shift algorithm', *The Journal of The Textile Institute*, Vol. 113, No. 1, pp.1–9.

Lee, J. and Hwang, K.I. (2022) 'YOLO with adaptive frame control for real-time object detection applications', *Multimedia Tools and Applications*, Vol. 81, No. 25, pp.36375–36396.

Li, Y., Chen, C.P. and Zhang, T. (2022) 'A survey on Siamese network: methodologies, applications, and opportunities', *IEEE Transactions on Artificial Intelligence*, Vol. 3, No. 6, pp.994–1014.

Liu, S., Huang, S., Xu, X., Lloret, J. and Muhammad, K. (2023) 'Efficient visual tracking based on fuzzy inference for intelligent transportation systems', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 12, pp.15795–15806.

Ma, L., Wang, Y.L. and Han, Q.L. (2022) 'Cooperative target tracking of multiple autonomous surface vehicles under switching interaction topologies', *IEEE/CAA Journal of Automatica Sinica*, Vol. 10, No. 3, pp.673–684.

Maghraby, Y.R., El-Shabasy, R.M., Ibrahim, A.H. and Azzazy, H.M.E.S. (2023) 'Enzyme immobilization technologies and industrial applications', *ACS Omega*, Vol. 8, No. 6, pp.5184–5196.

Oro, D., Fernández, C., Martorell, X. and Hernando, J. (2022) 'Work-efficient parallel non-maximum suppression kernels', *The Computer Journal*, Vol. 65, No. 4, pp.773–787.

Ram, S.S. (2022) 'Fusion of inverse synthetic aperture radar and camera images for automotive target tracking', *IEEE Journal of Selected Topics in Signal Processing*, Vol. 17, No. 2, pp.431–444.

Rani, R.S., Madhavan, P. and Prakash, A. (2022) 'Improving the mean shift clustering algorithm for universal background model (UBM)', *Circuits, Systems, and Signal Processing*, Vol. 41, No. 7, pp.3882–3902.

Terven, J., Córdova-Esparza, D.M. and Romero-González, J.A. (2023) 'A comprehensive review of yolo architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS', *Machine Learning and Knowledge Extraction*, Vol. 5, No. 4, pp.1680–1716.

Wang, K. and Liu, M. (2022) 'YOLOv3-MT: a YOLOv3 using multi-target tracking for vehicle visual detection', *Applied Intelligence*, Vol. 52, No. 2, pp.2070–2091.

Wu, W., Jin, X. and Tang, Y. (2020) 'Vision-based trajectory tracking control of quadrotors using super twisting sliding mode control', *Cyber-Physical Systems*, Vol. 6, No. 4, pp.207–230.

Yang, S. (2022) 'Face feature tracking algorithm of aerobics athletes based on Kalman Filter and mean shift', *International Journal of Biometrics*, Vol. 14, Nos. 3/4, pp.394–407.

Yang, W., Liu, H., Wang, Y. and Wang, X. (2023) 'Data-driven estimation of change-points with mean shift', *Journal of the Korean Statistical Society*, Vol. 52, No. 1, pp.130–153.

Zou, Q., Du, X., Liu, Y., Chen, H., Wang, Y. and Yu, J. (2022) 'Dynamic path planning and motion control of microrobotic swarms for mobile target tracking', *IEEE Transactions on Automation Science and Engineering*, Vol. 20, No. 4, pp.2454–2468.