

# Underground Coal Mine Target Tracking via Multi-Feature Joint Sparse Representation

Yan Lu, Qingxiang Huang

College of Energy Science and Engineering, Xi'an University of Science and Technology, Xi'an, China

Email: tyccly123@126.com

**How to cite this paper:** Lu, Y. and Huang, Q.X. (2021) Underground Coal Mine Target Tracking via Multi-Feature Joint Sparse Representation. *Journal of Computer and Communications*, 9, 118-132.  
<https://doi.org/10.4236/jcc.2021.93009>

**Received:** February 18, 2021

**Accepted:** March 28, 2021

**Published:** March 31, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Single-feature methods are unable to effectively track a target in an underground coal mine video due to the high background noise, low and uneven illumination, and drastic light fluctuation in the video. In this study, we propose an underground coal mine personnel target tracking method using multi-feature joint sparse representation. First, with a particle filter framework, the global and local multiple features of the target template and candidate particles are extracted. Second, each of the candidate particles is sparsely represented by a dictionary template, and reconstruction is achieved after solving the sparse coefficient. Last, the particle with the lowest reconstruction error is deemed the tracking result. To validate the effectiveness of the proposed algorithm, we compare the proposed method with three commonly employed tracking algorithms. The results show that the proposed method is able to reliably track the target in various scenarios, such as occlusion and illumination change, which generates better tracking results and validates the feasibility and effectiveness of the proposed method.

## Keywords

Underground Coal Mine, Sparse Representation, Particle Filter, Multi-Feature, Target-Tracking

## 1. Introduction

Video target tracking has been an important research topic in the field of computer vision. With the premise of the known target state in the first frame of a video, the task of tracking is to estimate the target state of subsequent frames by continuously measuring the target position. Target tracking has been applied in many fields [1] [2]. Applying visual tracking technology to underground coal mines has great significance for realizing coal mine automation and safety

monitoring. However, due to the uniqueness and complexity of the underground coal mine environment, the direct application of the existing tracking method to the coal mine environment is very difficult. First, an underground coal mine is a low-lit environment with uneven light distribution; second, the contrast between the underground coal mine target and the background is low; and third, when miners turn their bodies, the lamps on their helmets can cause a drastic change in the surrounding lighting of the target. These factors substantially affect the tracking performance of various methods [3].

Generative models and discriminative models are commonly employed in tracking models. Generative models treat the target tracking problem as a model matching problem and obtain the tracking result by searching for the candidate target region with the lowest reconstruction error [4], which is robust in the cases of occlusion and deformation during target tracking but encounters difficulties when attempting to maintain long-term tracking on targets with complex backgrounds. Discriminative models take into account information about both the target and the background and achieve tracking by classifying the target and the background using classifiers, which requires the extraction of positive and negative training samples to train and upgrade the classifiers during the tracking process. Reference [5] proposed an online multiple instance learning (MIL)-based target tracking algorithm. Reference [6] described a method in which after the tracking target is chosen, the recognition attributes of the target are utilized to establish a discriminant function to determine whether the area of the current frame belongs to the background or the foreground. Based on incremental learning, an incremental visual tracking (IVT) method was proposed by Ross [7] *et al.* using a particle filter framework. The compressive tracking (CT) algorithm proposed by Zhang [8] *et al.* employs compressed sensing to perform target representation in the compressed domain to perform real-time tracking. In the tracker  $l_1$  [9], by solving the problem of the minimum  $l_1$ , each candidate target is described by a sparse linear combination of a set of target templates, in which the corresponding reconstruction error is used to compute the observed probability of a candidate target. This method has demonstrated better tracking performance, especially in the case of noise and occlusion, but must find a solution for the minimum  $l_1$  problem for each candidate target, which is time-consuming. To improve its efficiency, Mei [10] introduced a sparse representation theory into the particle filter framework to track the target with minimum projection error; thus, the number of candidate targets whose sparse solutions need to be solved is decreased.

Target tracking in underground mine video has been rarely investigated, and tracking has been primarily performed based on electromagnetic localization [11] [12] [13]. Based on the Camshift algorithm, the combination of the local ternary patterns (LTP) texture model, the hue, saturation, value (HSV) model, and edge features, and the establishment of a feature contribution rule, Reference [14] proposed a real-time tracking method for an underground coal mine. Reference [15] argued that the tracking performance in a complex background

depends on the choice of the sample distribution and the establishment of an observation model and proposed the use of an unscented Kalman filter (UKF) as the recommend distribution and the combination of an optical flow histogram (representation of body motion) and a colour histogram as the observation model. However, when a colour histogram is used as the observation model, the tracking efficiency can decrease with an increase in the target size. The combination of an optical flow histogram and the distribution proposed by UKF can inevitably cause a further increase in the algorithm complexity, which produces uncertainty in the real-time performance. We propose a target tracking algorithm that is suitable for an underground coal mine environment in this paper. Multi-feature extractions are performed on a target template and candidate target image, based on the principle of sparse representation, to improve the robustness of the sample features. The colour feature and texture feature of the sample in the observation model are fused to generate the fusion feature model to adapt to uneven and drastic changes in the underground illumination, and combine the joint appearance model with the particle filter framework. The tracking result is obtained using the sparse representation theory. Experiments with videos from the standard video library validate the advantages of the proposed method in terms of accuracy, stability, real-time performance, and adaptability to the underground coal mine environment [3].

## 2. Basic Principle

### 2.1. Colour Feature

Due to various issues in underground mine video, such as uneven illumination, drastic light changes, and noise interference, we combined the HSV colour feature [16] with the scale invariant local ternary pattern (SILTP) texture feature [17] to describe the moving target to improve the robustness of the target template. Two feature types are described in this section.

Hue (H), saturation (S), and value (V) are the three components of The HSV colour space [16]. Compared with the three feature components of the red, green, and blue (RGB) colour space, the three feature components of the HSV colour space are uncorrelated and more consistent with the visual characteristics of the human eye. Therefore, the use of the HSV space as a colour feature model can mitigate the interference of lighting. The HSV colour histogram feature extraction method is described as follows [18]:

#### 1) Colour space conversion

According to Equation (1), the image can convert from the RGB to the HSV:

$$\begin{cases} V = \frac{1}{\sqrt{3}}[R + G + B] \\ S = 1 - \frac{\sqrt{3}}{V} \min(R, G, B) \\ H = \begin{cases} \theta & G \leq B \\ 2\pi - \theta & G < B \end{cases} \end{cases} \quad (1)$$

and  $\theta$  is the angle.

$$\theta = \cos^{-1} \left[ \frac{1/2[(R-G)+(R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right] \quad (2)$$

## 2) Conversion of the HSV colour space and histogram

The HSV colour space is uniformly quantized to a 256-dimension histogram using the uniformly spaced quantization method, in which the colour quantities of the  $H$ ,  $S$ , and  $V$  were uniformly quantized to 16 parts, 4 parts, and 4 parts, respectively, *i.e.*, the quantization levels of the three quantities are 16, 4, and 4, respectively.

3) The 256-dimension histogram is encoded using the Haar transform method, and the features of the encoded histogram are the desired HSV colour space-based features. As shown in **Figure 1**.

## 2.2. Texture Features

The SILTP feature is a modified local binary pattern (LBP) description operator with excellent robustness to illumination variation and noise; thus, it can effectively solve the light-derived histogram instability issue as it has a better adaptability to sudden lighting changes. The SILTP feature has excellent robustness to area-wide noise, especially shadows and local noise; thus, it is suitable for extracting features from underground mine video. The SILTP operator uses 00, 01, and 10 to represent the pixel texture, and increases it by one digit compared with the LBP operator. Assume that the position of the centre pixel of the image block is  $(x_c, y_c)$  and the grey value of the centre pixel is  $I_c$ . The encoding function is defined as

$$s_\tau(I_c, I_k) = \begin{cases} 01 & I_k > (1+\tau)I_c \\ 10 & I_k < (1-\tau)I_c \\ 00 & \text{others} \end{cases} \quad (3)$$

$$SILTP_{Q,R}^\tau(x_c, y_c) = \bigoplus_{k=0}^{Q-1} s_\tau(I_c, I_k) \quad (4)$$

The grey value of the  $Q$  neighbourhood pixel with a threshold of  $\tau$  and a

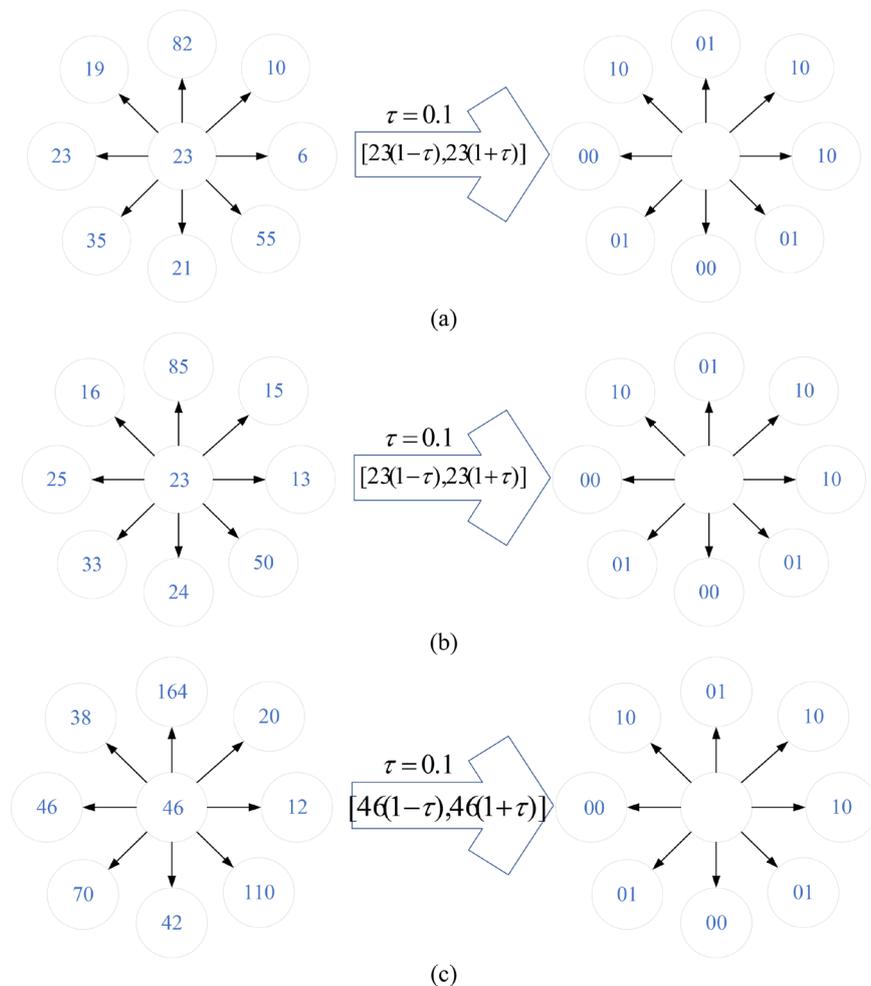


**Figure 1.** Comparison of the RGB space picture and HSV space picture. (a) RGB space picture; (b) HSV space picture.

radius of  $R$  is  $I_k$ .

The binary values of the neighbourhoods of the centre pixel are connected in a counter clockwise direction by  $\oplus$  to form a string

The original encoding process is shown in **Figure 2(a)**; in the case of local noise in the environment, as shown in **Figure 2(b)**, the noise within a certain range did not affect the encoding result. When the light changes, the light intensity is doubled, and the grey value of each pixel and the grey values of the centre point and the neighbourhood are doubled (**Figure 2(c)**), which causes no changes in the coding result. Therefore, the SILTP has excellent robustness to illumination change and local noise. The advantages of the SILTP operator are reflected in the following three aspects. First, for each neighbourhood, only one more comparison is needed relative to the LBP operator; thus, the computational efficiency is very high. Second, the SILTP operator is very robust to region-wide noise; especially when the detection region is darker or contains more noises, SILTP is more adaptive to these changes than LTP or LBP and generates better inspection results. Last, the scale invariance enables the SILTP operator to be



**Figure 2.** SILTP encoding process. (a) Original encoding process; (b) Encoding process in the presence of noise; (c) Encoding process in the case of doubled grey value of the pixel.

more robust to brightness change, and even in the case in which the light suddenly changes from dark to bright, the SILTP features do not change. The operator is also able to recognize when the background area is in a weak shadow, as the weak shadow retains the texture information of the dark background rather than the texture information represented by the scale factor of the local background area [19].

### 2.3. Particle Filtering

A particle filter is a statistical filter method based on Monte Carlo and recursive Bayesian estimation [20]. Recursive Bayesian estimation refers to the process in which the prior distribution is modified by adjusting the weights and positions of the particles based on the posterior probability distribution recursive from the existing prior probability distribution. The key idea of a particle filter is to extract a set of random weighted samples  $(\{x_{k-1}^i\}, i=1, 2, \dots, N)$  from the prior probability distribution to approximately represent the posterior probability density function. Thus, the integral operation in the derivation process is converted to the summation operation of finite sample points. The approximation equation of the posterior density function is

$$p(x_k | z_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(x_k - x_k^i) \quad (5)$$

where  $\delta(x)$  is the Dirac function;  $x_k$  is the state value at time  $k$ ; and  $z_{1:k}$  is the observed values from time 1 to time  $k$ . Performing sampling from the posterior probability density function is often difficult, and generally, sampling is performed from an alternative significance density function  $q(x_k^i | x_{k-1}^i, z_k)$  as the sampling by this approach can be readily performed. The update equation of particle weight in the recursive process is expressed as follows:

$$\omega_k^i = \omega_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)} \quad (6)$$

After normalizing the weights, the status output is expressed as follows:

$$x_k = \sum_{i=1}^N \omega_k^i x_k^i \quad (7)$$

### 2.4. Particle Filter Based on the Fruit Fly Optimization Algorithm (FOA)

Derived from fruit fly foraging behaviour, the FOA is a global optimization-pursuing swarm intelligence algorithm [21]. Based on characteristic fruit fly foraging behaviours, the FOA consists of the following three steps [22]:

1) Population initialization: the number of fruit fly populations ( $N$ ), the maximum number of iterations ( $m$ ), and the initialized foraging locations of an individual fruit fly ( $X_0, Y_0$ ) are determined.

2) Foraging activities: starting from the initial position, the fly swarm uses the olfactory sense to search for food, and the search direction and distance of each

individual fruit fly are random numbers with a set range. The fitness value of each individual fruit fly  $f_i$  (odour concentration judgement value) is calculated based on its current position  $(X_i, Y_i)$ , and the individual with the highest fitness value is determined. The swarm flies towards the position of this individual.

3) Update the swarm position: in each iteration, if the next optimal fitness value is greater than the prior optimal fitness value, the swarm updates its position of the optimal fitness value as a new foraging position  $(X_0, Y_0)$  towards which all individuals fly and then fly out to continue the foraging. This process is repeatedly iterated until the maximum number of iterations or the set accuracy is attained.

The resampling process of the traditional particle filter algorithm solves the particle degeneracy problem by duplicating large-weight particles and deleting small-weight particles. However, after multiple iterations, the particle depletion issue is observed. Specifically, the rationale of the fruit fly optimization particle filter algorithm is described as follows. In the standard particle filter process, after obtaining  $N$  particles by importance sampling, the fitness value of each particle is calculated based on the position of the particle and the fitness function. If the particle set is distributed in the vicinity of the true state, then the fitness value of each particle in the swarm is high; conversely, the particle set is not distributed in the vicinity of the true state. Optimization with the particle distribution by the FOA enables the particles to constantly fly towards the position with a high fitness value to ensure that the particles are constantly getting close to the true state, which improves the quality of the total samples of the particle swarm.

The FOA optimizes and improves the resampling process of particle filtering, in which the signal point particles are represented by the fruit fly swarm, and the more effective particles are obtained via the high optimization capability of the FOA. Thus, the particle depletion problem is solved.

By multiple iterations, the FOA updates the positions of fruit fly individuals, and the individual fruit fly with the highest odour concentration judgement value is deemed the optimal solution. The position update is implemented using the following equations:

$$X_{m+1}^i = X_m^i + 2r \times \text{randomvalue} - r \quad (8)$$

$$Y_{m+1}^i = Y_m^i + 2r \times \text{randomvalue} - r \quad (9)$$

The odour concentration value is controlled by the following equation:

$$M_k^i = \exp\left\{-\frac{1}{2R_k}(z_n - z_p)\right\} \quad (10)$$

where  $m$  is the number of iterations, and  $i = 1, 2, \dots, N$  is fruit fly individuals;  $N$  is the total number of fruit fly individuals;  $\text{randomvalue} - r$  is a random number in the range of  $[0, 1]$ ;  $R_k$  is the measurement noise variance;  $z_n$  is the updated measurement value, *i.e.*, the actual distance from the particle to the target; and  $z_p$  is the predicted measurement value, *i.e.*, the predicted distance from the particle to the fruit fly individual.

The process of the FOA-optimized particle filter resampling is described as follows:

- 1) Initialization: let  $m$  in Equations (8) and (9) be  $m = 0, i = 1, 2, \dots, N$ ;
- 2) Olfactory search: update paths according to Equations (8) and (9) to search the particle at the signal point with the largest weight;
- 3) Evaluation of individuals: calculate the concentration of the odour released by an individual fruit fly ( $M_k^i$ ) using Equation (10);
- 4) Visual search: choose the individual fruit fly with a  $\max\{M_k^i\}$  as the particle with a  $\max\{\omega_k^i\}$ ;
- 5) When the termination criteria are satisfied, the particle with the maximum weight is outputted. Otherwise, let  $m = m + 1$ , go back to Step (2) [23].

### 3. Sparse Representation

#### 3.1. Sparse Representation

Sparse representation is using a linear superposition of a base function dictionary to represent the natural images or signals [24]. For a given signal  $Y \in R^n$  and a given base function dictionary  $D = \{d_1, d_2, \dots, d_n\} \in R^{m \times n}$ , the sparse model of  $Y$  can be expressed as

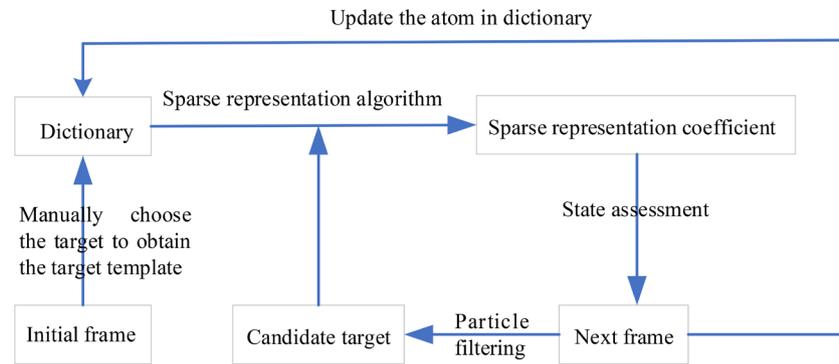
$$\min \|x\|_0 \quad \text{s.t. } Y = Dx \quad (11)$$

where  $x$  is the sparse representation of signal  $Y$  in dictionary  $D$  and  $\|x\|_0$  is the number of non-zero components in  $x$ . When the number of dimensions of dictionary  $D$  satisfies the condition of  $m > n$ , the linear equation  $Y = Dx$  is an over determined condition, and a unique solution for  $x$  exists. When  $m < n$ ,  $Y = Dx$  is an underdetermined equation, and the solution of  $x$  is not unique. Certain constraints need to be added; thus, solving Equation (11) is converted to the optimization of Equation (12).

$$\min \|x\|_0 \quad \text{s.t. } \|Y - Dx\|_2^2 \leq \varepsilon \quad (12)$$

Equation (12) is a sparse model with the condition that  $Y$  satisfies the approximation error, where  $\|x\|_0$  represents the number of non-zero components in  $x$ , *i.e.*, the sparsity of the matrix  $x$ , and  $\varepsilon$  represents the error. The main framework of the sparse representation-based target tracking algorithm is shown in **Figure 3**. The target template is manually chosen in the initial frame, and the target template dictionary is constructed. When tracking the current frame, particle sampling is performed on the current frame to obtain the candidate target. The sparse representation coefficient of each candidate target in the dictionary is calculated using the sparse representation method. The state of the target in the current frame is assessed based on the sparse representation coefficient, and thus, using the target in the current frame to update the dictionary.

However, directly solving the norm of  $l_0$  in Equation (12) is an NP-hard problem, which is generally solved by the approximation method. The algorithms for sparse solution can be approximately categorized into three categories:



**Figure 3.** Flow diagram of the target tracking algorithm via sparse representation.

### 1) Greedy algorithm

The core idea of the greedy algorithm is that the maximum value of the inner product of the residual of the previous step and the optimal atom is chosen in each step. For the minimization of the  $l_0$  norm, the solutions that use the greedy algorithm include the matching pursuit (MP) algorithm [25], which represents the first greedy algorithm, the orthogonal matching pursuit (OMP) algorithm [26] and its modified algorithms [27] [28]. The OMP algorithm uses orthogonal projection transformation to project the base vector into the subspace and then calculates the residual of the representation, which is iterated until the sparse coefficient of the original signal is obtained. Compared with the MP algorithm, the computational efficiency of the OMP algorithm has been significantly improved. The MP algorithm is adopted in this study to solve Equation (12).

### 2) Convex relaxation algorithm

The convex relaxation algorithm relaxes the minimization problem of the norm  $l_1$  into the minimization problem of the norm  $l_1$  with restricted isometry property (RIP) conditions. Chen [29] *et al.* proposed the basis pursuit (BP) algorithm, which employs the interior point method for linear programming to solve the minimization problem of the norm  $l_1$ , which represents the optimization method by solving the approximated norm of the norm  $l_1$ .

In this study, the OMP algorithm is used to solve the sparse coefficient of Equation (12).

## 3.2. Dictionary Update

During the tracking process, affected by various factors, such as posture change, brightness change, and occlusion, the appearance of the target may change. Thus, the target template set should be updated to adapt to changes in the appearance of the target. In this study, the following update strategy is adopted:

- 1) The target that is manually selected in the first frame to be unsubstituted, *i.e.*, the first component of the target template set, is maintained;
- 2) Starting from the second frame, after determining the target  $\hat{y}$ , the probability of the similarity between the target template set and  $\hat{y}$  is evaluated. If the probability is lower than a given threshold, then the poorest target template

is replaced with  $\hat{y}$  [30].

#### 4. Proposed Algorithm

Step 1: Initialize.

- 1) Manually select the moving target in the first frame  $y_0$ ;
- 2) Collect a set of samples in the vicinity of the initial state  $y_0$  via a Gaussian distribution, establish the target template set  $T = \{T_1, T_2, \dots, T_N\}$ , and extract the HSV and SILTP features of each target template in the target template set to obtain the target template set ( $T^k$ ) of each modal feature.

Step 2: Starting from the second frame, continue the following steps to the end of the video.

- 1) In frame  $t$ , sample  $N$  candidate targets ( $\hat{y}_{(t,1)}, \hat{y}_{(t,2)}, \dots, \hat{y}_{(t,N)}$ ) in the vicinity of the target location ( $\hat{y}_{t-1}$ ) of the previous frame using the particle filter method
- 2) Extract the HSV and SILTP features of the candidate targets ( $\hat{y}_{(t,i)}, i = 1, 2, \dots, N$ ) to obtain the corresponding feature vector representation  $\hat{y}_{(t,i)}^k$ .
- 3) Calculate the sparse coefficient and the reconstruction residual using the OMP algorithm, and determine the tracking target based on the sparse representation reconstruction residual, *i.e.*, the candidate target with the smallest reconstruction residual is the desired target.
- 4) Calculate the probability of each candidate target ( $\hat{y}_{(t,i)}, i = 1, 2, \dots, N$ ) and the target template set.
- 5) Update the target template set according to the update rule.
- 6) Perform resampling to generate  $N$  particles in the next frame to be tracked.
- 7) Repeat Step 2 to the end of the video.

#### 5. Experiment and Analysis

The algorithms in this experiment are implemented on a PC with 2.50-GHz Intel Core i7-4710HQ processor and 8 GB of memory and the MATLAB 2011b compiler environment. For each test video segment, the target location in the first frame was manually labelled, and the number of templates of dictionary  $D$  is 20 (*i.e.*,  $m = 20$ ). The number of particles is set to 300 (*i.e.*,  $n = 300$ ), and the dictionary is updated as previously described. The algorithm proposed in this paper compared with the multiple target tracking (MTT) algorithm, the IVT algorithm, and the  $l_1$  algorithm.

##### Qualitative Analysis

We compared the four algorithms using four video segments in the experiments; the experimental results are shown in **Figure 4**, **Figure 5**, **Figure 6**, **Figure 7**. In the first video segment, the problems of retention and re-movement of moving target are present, and all algorithms are able to track the moving target. However, the tracking boxes of the IVT and  $l_1$  algorithms are not very accurate, while the proposed method and the MTT algorithm can accurately track the target. In



Figure 4. Tracking results on the first video segment. (a) MTT; (b) IVT; (c)  $I_1$ ; (d) Proposed method.



Figure 5. Tracking results for Frames 122 and 149 of the second video segment. (a) MTT; (b) IVT; (c)  $I_1$ ; (d) Proposed method.



Figure 6. Tracking results for the third video segment. (a) MTT; (b) IVT; (c)  $I_1$ ; (d) Proposed method.



Figure 7. Tracking results for the fourth video segment. (a) MTT; (b) IVT; (c)  $I_1$ ; (d) Proposed method.

the second video segment, blurry images and drastic illumination change are present, the tracking results indicated that the MTT and  $I_1$  algorithms are completely unable to track the moving target, with tracking failure, and the IVT algorithm shows a high tolerance to a change in illumination. In the case of drastic changes in illumination, however, the tracking box gradually deviates from the moving target, while the proposed algorithm can effectively track the moving target. In the third video segment, the effect of a change in ambient light on the moving target is not as profound as that in the second video segment. In addi-

tion to the IVT algorithm (its tracking box is drifting), the remaining algorithms are able to track the moving target very well. The fourth video segment is the live footage of a miner working in a mine, in which the lamp on the miner's helmet causes a dramatic change in light around the target. Due to the effect of a change in light, the MTT algorithm completely lost the tracking target, and the  $I_1$  algorithm shows the tracking drift but is able to track the target. Although the IVT algorithm is robust to a change in illumination, its tracking box significantly drifts, while the proposed algorithm shows a strong robustness to the illumination change, and its tracking box is able to accurately track the moving target.

Based on these results, in the cases of weak and strong changes in light around the target, the proposed algorithm can effectively track the moving target and demonstrate better robustness and adaptability in solving the tracking problems for different scenes.

Based on the average centre position error, we evaluate the experimental results of the tracking algorithms; the lower the centre position error is, the more accurate the obtained tracking result and the better the algorithm performance, and vice versa.

**Table 1** shows that the average centre position error of the proposed algorithm for the four video segments is 4.875 pixels, which is considerably superior to that of the second best IVT algorithm (23.625 pixels). According to the statistical results, the proposed algorithm shows better tracking stability and is more adaptive to the unique underground mine environment with illumination changes than the other algorithms. The algorithm proposed in paper has advances the other algorithms.

## 6. Conclusion

In this study, we propose a tracking algorithm via multi-feature joint sparse representation for the underground mine environment. The global HSV features and local SILTP features are combined based on the characteristics of the underground mine video, sparse representation is performed on a moving target via a particle filter framework, and the established target is updated in real time to better adapt to changes in the features of the target. We compare the proposed

**Table 1.** Average centre position error (unit: pixel).

<i>Video Segment</i>	<i>MTT</i>	<i>IVT</i>	$I_1$	<i>Proposed algorithm</i>
1	3.6	7.4	4.9	2.1
2	148.4	45.1	165.2	9.6
3	2.3	23.4	3.5	2.5
4	119.2	18.6	25.2	5.3
Mean	68.375	23.625	49.7	4.875

method with three mainstream tracking algorithms using four video segments with different scenarios, analyse the performance and accuracy of the proposed method from two perspectives, and conclude that the proposed algorithm is able to overcome a variety of tracking obstacles in different environments to achieve stable tracking. Compared with the other algorithms, the algorithm proposed in this paper shows higher accuracy and tracking robustness.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Shu, C.F., Hampapur, A., Lu, M., *et al.* (2005) IBM Smart Surveillance System (S3): An Open and Extensible Framework for Event Based Surveillance. *IEEE Conference on Advanced Video and Signal Based Surveillance*, Como, 15-16 September 2005, 318-323.
- [2] Shah, M., Javed, O. and Shafique, K. (2007) Automated Visual Surveillance in Realistic Scenarios. *IEEE Multimedia*, **14**, 30-39. <https://doi.org/10.1109/MMUL.2007.3>
- [3] Sun, J.P. (2015) Personnel Target Matching and Tracking Method in Mine Video Images. *Journal of China University of Mining and Technology*, **44**, 540-548.
- [4] Mao, Z.C. and Huang, S.W. (2018) Target Tracking Based on Structured Discriminant Sparse Representation. *Journal of Nanjing University of Science and Technology*, **42**, 271-277.
- [5] Babenko, B., Yang, M. and Belongie, S. (2011) Robust Object Tracking with Online Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 1619-1632. <https://doi.org/10.1109/TPAMI.2010.226>
- [6] Xu, J. and Wang, X.F. (2013) Recognition Method of Moving Target Using Bayesian Probability Theory. *Journal of Nanjing University of Science and Technology*, **37**, 80-84.
- [7] Ross, D.A., Lim, J., Lin, R.S., *et al.* (2008) Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, **77**, 125-141. <https://doi.org/10.1007/s11263-007-0075-7>
- [8] Zhang, K., Zhang, L. and Yang, M.H. (2012) Real-Time Compressive Tracking. *European Conference on Computer Vision*, Florence, 7-13 October 2012, 864-877. [https://doi.org/10.1007/978-3-642-33712-3\\_62](https://doi.org/10.1007/978-3-642-33712-3_62)
- [9] Mei, X., Ling, H.B., Wu, Y., *et al.* (2011) Minimum Error Bounded Efficient  $l_1$  Tracker with Occlusion Detection. 2011 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, 20-25 June 2011, 1257-1264. <https://doi.org/10.1109/CVPR.2011.5995421>
- [10] Mei, X. and Ling, H.B. (2011) Robust Visual Tracking and Vehicle Classification via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 2259-2272. <https://doi.org/10.1109/TPAMI.2011.66>
- [11] Sun, J.P. and Wang, S. (2014) Application of the Improved Energy Transfer Ranging Model in Mine Localization. *Journal of China University of Mining & Technology*, **43**, 94-98.
- [12] Sun, J.P. and Li, C.X. (2013) In-Pit Coal Mine Personnel Uniqueness Detection

- Technology Based on Personnel Positioning and Face Recognition. *International Journal of Mining Science and Technology*, **23**, 357-361.  
<https://doi.org/10.1016/j.ijmst.2013.05.014>
- [13] Su, J.P. and Li, C.X. (2014) Mine TOA Positioning Method Based on Kalman Filtering and Fingerprint Positioning. *Journal of China University of Mining & Technology*, **43**, 1127-1133.
- [14] Li, D., Tian, J., Xiao, L.Q., *et al.* (2013) Adaptive Multi-Feature Template Video Target Tracking Based on Camshift Algorithm. *Journal of China Coal Society*, **38**, 1299-1304.
- [15] Tian, J., Qian, J.S., Li, D., *et al.* (2011) Unscented Particle Filter Algorithm for People Tracking in Coal Mines Based on Adaptive Multi-Cues Integration Models. *Journal of China University of Mining & Technology*, **40**, 146-151.
- [16] Wang, Q., Feng, D.D. and Chi, Z. (2004) B-Spline Over-Completes Wavelet Based Fractal Signature Analysis for Texture Image Retrieval. *International Symposium on Intelligent Multimedia*, Hong Kong, 230-242.
- [17] Liao, S.C., Zhao, G.Y., Kellokumpu, V., *et al.* (2010) Modeling Pixel Process with Scale Invariant Local Patterns for Background Subtraction in Complex Scenes. *Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, 13-18 June 2010, 1301-1306.  
<https://doi.org/10.1109/CVPR.2010.5539817>
- [18] Li, D. and Peng, Y.G. (2016) Eye Fatigue Diagnosis Method Based on Feature Fusion by HSV and LBP. *Acta Automatica Sinica*, **37**, 77-82.
- [19] Yang, G.L. (2014) Moving Object Detection Algorithm Using SILTP Texture Information. *Computer Science*, **4**, 302-305.
- [20] Zhu, Z.Y. (2010) Particle Filter Algorithm and Its Application. Science Press, Beijing, 27-31.
- [21] Pan, W.T. (2012) A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example. *Knowledge Based Systems*, **26**, 69-74.  
<https://doi.org/10.1016/j.knosys.2011.07.001>
- [22] Han, K. and Zhang, H. (2018) Improved Particle Filter Based on Fruit Fly Optimization and Its Application in Target Tracking. *Journal of Hunan University (Natural Science)*, **45**, 130-138.
- [23] Fu, H. and Qi, X.J. (2019) Research on De-Noising of AE Signals of Coal-Gas Outbursts by Improved PF Algorithm. *Application Research of Computers*, **36**, 720-723.
- [24] Lian, Q.S. and Chen, S.Z. (2010) Sparse Image Representation Using the Analytic Contourlet Transform and Its Application on Compressed Sensing. *Acta Electronica Sinica*, **38**, 1293-1298.
- [25] Mallat, S. and Zhang, Z. (1993) Matching Pursuit with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, **41**, 3397-3415.  
<https://doi.org/10.1109/78.258082>
- [26] Donoho, D.L., Tsai, Y., Drori, I., *et al.* (2012) Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, **58**, 1094-1121.  
<https://doi.org/10.1109/TIT.2011.2173241>
- [27] Sun, L., Ji, S. and Ye, J. (2011) Canonical Correlation Analysis for Multilabel Classification: A Least-Squares Formulation, Extensions, and Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**, 194-200.

<https://doi.org/10.1109/TPAMI.2010.160>

- [28] Dutta, S. and De, A. (2014) LAMP: A Locally Adapting Matching Pursuit Framework for Group Sparse Signatures in Ultra-Wide Band Radar Imaging.
- [29] Chen, S.S., Donoho, D.L. and Saunders, M.A. (1998) Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, **20**, 33-61.  
<https://doi.org/10.1137/S1064827596304010>
- [30] Duan, X.P., Liu, J.F., Wang, J.H., *et al.* (2015) Video Target Tracking via Multi-Cue Joint Sparse Representation. *Journal of Harbin Engineering University*, **36**, 1609-1913.