applied sciences

MDPI

Article

Tracking System for a Coal Mine Drilling Robot for Low-Illumination Environments

Shaoze You [1,2], Hua Zhu [1,*], Menggang Li [1], Yutan Li [1,3] and Chaoquan Tang [1]

[1] School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou 221116, China
[2] Jiangsu Collaborative Innovation Center of Intelligent Mining Equipment, China University of Mining and Technology, Xuzhou 221008, China
[3] School of Intelligent Manufacturing, Jiangsu Vocational Institute of Architectural Technology, Xuzhou 221116, China
* Correspondence: zhuhua83591917@163.com; Tel.: +86-516-83591917

Abstract: In recent years, discriminative correlation filters (DCF) based trackers have been widely used in mobile robots due to their efficiency. However, underground coal mines are typically a low illumination environment, and tracking in this environment is a challenging problem that has not been adequately addressed in the literature. Thus, this paper proposes a Low-illumination Long-term Correlation Tracker (LLCT) and designs a visual tracking system for coal mine drilling robots. A low-illumination tracking framework combining image enhancement strategies and long-time tracking is proposed. A long-term memory correlation filter tracker with an interval update strategy is utilized. In addition, a local area illumination detection method is proposed to prevent the failure of the enhancement algorithm due to local over-exposure. A convenient image enhancement method is proposed to boost efficiency. Extensive experiments on popular object tracking benchmark datasets demonstrate that the proposed tracker significantly outperforms the baseline trackers, achieving high real-time performance. The tracker's performance is verified on an underground drilling robot in a coal mine. The results of the field experiment demonstrate that the performance of the novel tracking framework is better than that of state-of-the-art trackers in low-illumination environments.

Keywords: visual object tracking; low illumination; image enhancement; computer vision; mobile drilling robot; coal mine robot

Citation: You, S.; Zhu, H.; Li, M.; Li, Y.; Tang, C. Tracking System for a Coal Mine Drilling Robot for Low-Illumination Environments. *Appl. Sci.* **2023**, *13*, 568. https://doi.org/10.3390/app13010568

Academic Editor: Alessandro Gasparetto

Received: 18 October 2022
Revised: 22 December 2022
Accepted: 29 December 2022
Published: 31 December 2022

1. Introduction

Visual object tracking (VOT) is a fundamental problem and a popular research area in computer vision. Numerous studies have been conducted on this topic, producing multiple performance evaluation datasets and benchmarks [1–5]. The goal of the tracker is to select a model-free target from the first frame and track it in subsequent frames of a video stream or image sequence. Recently, VOT has been used for real-time vision applications, such as intelligent monitoring systems, automatic driving systems, and robotics [6–8].

Object tracking is an online task that needs to meet the requirements of practical applications. An ideal tracker should be accurate and robust for a long period in real-time vision systems. Moreover, due to the complexity of the working environment, including image deformation, object occlusion, illumination changes, motion blur, and objects out of view, the tracker is prone to drifting during long-term tracking, reducing its performance [9]. In recent years, DCF-based methods, including Minimum Output Sum of Squared Error (MOSSE) [10], kernelized correlation filter (KCF) [11], background-aware correlation filter (BACF) [12], discriminative correlation filter with channel and spatial reliability (CSR-DCF) [13], and efficient convolution operators (ECO) [14], have significantly advanced the state-of-the-art (SOTA) performance for short-term tracking. These methods have high processing speeds and are convenient for feature extraction, resulting in a new research direction in this field.

Intelligent mobile robots have become a research hotspot in science and technology. Path planning, positioning and navigation, obstacle avoidance, and other aspects of mobile robots are inseparable from the assistance of vision technology [15]. Since mobile robots are limited by their endurance, volume, and flexibility, their hardware configuration is usually based on low power consumption. Thus, an algorithm with high computational complexity can substantially reduce the real-time performance of the robot. In practical engineering applications, intermittent light sources or uneven illumination will cause overexposure of the camera, which can adversely affect the performance of VOT. This problem is particularly pronounced in coal mines. The distribution form of a miner's lamp cannot make the light fill the whole tunnel uniformly (As shown in Figure 1). Most coal mines use infrared cameras to deal with low illumination, but these images are single-channel images (grayscale images). Infrared cameras are typically used to monitor fixed equipment, resulting in limitations in dynamic object tracking, such as personnel monitoring and mining vehicle scheduling. Relevant research shows that the performance of the vision algorithm is directly related to the richness of image features [16]. The performance of current trackers is affected by the number of characteristic channels. It is necessary to analyze the performance of target trackers using a color camera in low-illumination environments. Therefore, an algorithm to track dynamic targets in low-illumination environments with a color image is required. Moreover, this algorithm can also need to be applied to mobile robots with power consumption constraints.

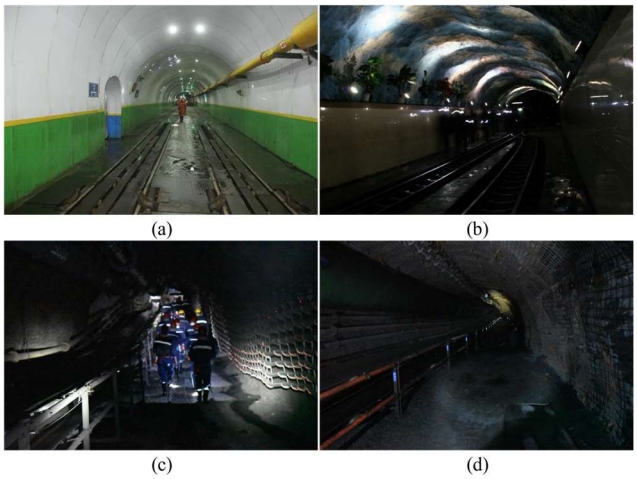

Figure 1. Typical environments in underground coal mine. (**a**) is a main roadway of a coal mine. (**b**) is the shaft bottom of coal mine. (**c**) is general roadway in coal mine. (**d**) is a haulage roadway.

Hence, this work focuses on practical problems in engineering applications rather than dataset evaluations. We propose the low-illumination long-term correlation tracker (LLCT). It is adapted to a low-illumination environment, outperforms SOTA trackers of the time, and provides real-time and high-precision performance in the field environment. The LLCT uses an image exposure compensation method for low-illumination environments, making it suitable for coal mine mobile robots. The flowchart of the proposed method is illustrated in Figure 2.

The long-term real-time correlation filter (LRCF) for mobile robots proposed in [17] has been significantly improved and extended in this paper. The contributions of our previous study are summarized as follows:

1. Principal component analysis (PCA) was used with the LRCF to reduce the dimensionality in the translation and scale estimation phase to improve the algorithm speed.
2. The memory templates were updated at regular intervals, and the existing and initial templates were re-matched every few frames to maintain template accuracy.

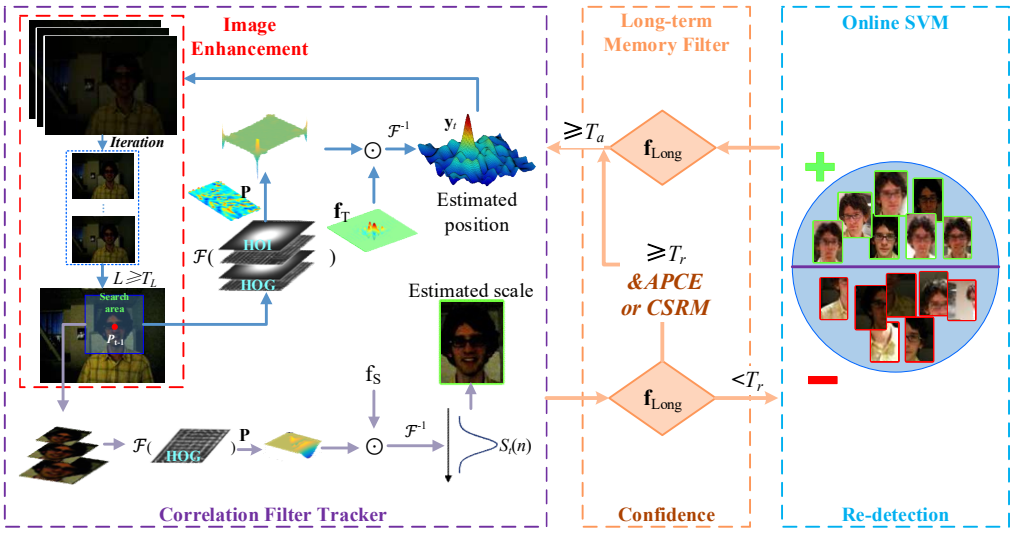

Figure 2. The flowchart of the proposed tracking method. The projection matrix **P** was applied to the LCT [18] architecture to reduce the computational complexity. The need for image enhancement is evaluated before sample training. Image enhancement is performed iteratively until the image reaches the brightness threshold.

The contribution of this study and the improvements are as follows:

1. This work designs a target tracker framework in a low-illumination coal mine environment. The environmental illumination is detected before the DCF extracts the training sample. An image enhancement module is incorporated.
2. A local illumination detection method is proposed. A pre-set padding area around the target bounding box (BB) is cropped and used as the area of local illumination detection. The illumination intensity in this area, instead of the global illumination, determines whether image enhancement is performed.
3. A fast and efficient image enhancement method is proposed because most image enhancement algorithms are computationally expensive and have low real-time performance. Excellent image enhancement algorithms bring expensive computational costs and perform poorly in real-time performance. This method is well-suited to object tracking by robots.
4. A tracking system based on a robot operating system (ROS) is designed. The proposed tracker installed in a drilling robot is evaluated in a field experiment conducted in an underground coal mine.

The remainder of this paper is organized as follows. Section 2 discusses previous work related to the proposed trackers. The proposed DCF tracking method and the principle of the image enhancement module are introduced in Section 3. The experimental results are analyzed and discussed in Section 4. Finally, the conclusion is presented in Section 5.

2. Related Works

In this section, we review the recent achievements related to our proposed approach. For a comprehensive overview of existing tracking methods, readers can refer to the cited articles.

The tracking-by-detection method regards the target tracking in each frame as a detection problem in a local search window. The method usually separates the target from its surrounding background by an incremental learning classifier [11,19–21]. In this kind of study, each frame in the image sequence is regarded as a single target detection process. This approach is currently the most commonly used method for visual target tracking.

In 2011, Bolme et al. proposed the MOSSE [10] filter for tracking, which exhibited an impressive speed of more than 600 frames per second (FPS), demonstrating the potential

of the correlation filter. DCF-based trackers use fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT) for learning and rapid detection in the frequency domain. Thus, they have a low computing time by means of correlation operation on image features. Many DCF-based trackers have been proposed for feature extraction with a learning filter architecture [10–14,20,22]. However, most sacrifice the tracking speed to achieve accurate and robust tracking performance.

In recent years, deep learning (DL) has been widely utilized in computer vision, language processing, and intelligent robotics. As a representative architecture, convolutional neural networks (CNNs) have achieved remarkable results in visual tracking due to their powerful feature expression ability. Three types of CNNs have been used for VOT: (a) CNNs based on pure convolutional features [23]; (b) Siamese network-based trackers [21,24,25]; (c) DCF-based trackers based on the VGG-Net [26] and other network training features [22,27,28]. These trackers provide high-precision results by utilizing the graphics card. However, they require many training samples and high computational power, limiting their application on mobile devices.

Long-term tracking differs from short-term tracking because of long-term occlusion and field-of-view conditions. Zdenek et al. [29] first proposed this tracking framework and decomposed the long-term tracking task into three parts: tracking, learning, and detection (TLD). The long-term correlation tracker (LCT) [18] and LCT+ [30] were proposed by Ma et al. in the framework of the convolution kernel correlation filter tracker. A re-detector and a long-time filter were added to the general short-term tracker architecture. Zhu et al. proposed a novel collaborative correlation tracker (CCT) [31] using multi-scale kernelized correlation tracking (MKC) and an online CUR [32] filter for long-term tracking. Yan et al. [33] proposed a Skimming-Perusal tracking framework, which is based on deep networks and SiameseRPN [34] to achieve real-time and robust long-term tracking works.

Low illumination tracking task has been a key concern in coal mine application, Shang et al. [35] prefer to use a Kinect camera for low illumination tracking task. Li et al. [36] proposed a dual correlation filtering structure for tracking in dark light, which is called an anti-dark tracker (ADTrack). Ye et al. [37] tended to use a CNNs-based framework to realize reliable UAV tracking at night and proposed a spatial-channel transformer-based low-light enhancer (SCT) [38].

3. The Proposed Method

3.1. Discriminative Correlation Filter

The fast DSST [39] is adopted as the baseline due to its outstanding performance. It utilizes two optional correlation filters for estimating the translation (two-dimensional) and scale (one-dimensional) of the target in the new frame. The goal for a two-dimensional (M × N pixels) image is to learn a set of multichannel correlation filters $\mathbf{f}_t^l \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}\times\mathrm{D}}$ based on the sample $\{(\mathbf{x}_t, \mathbf{y}_t)\}_t^l$ at the time t. Each training sample $\mathbf{x}_t \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}\times\mathrm{D}}$ contains D-dimensional features extracted from the interest region, and the features channel $l \in \{1, \ldots, D\}$ of $\mathbf{x}_t$ is denoted as $\mathbf{x}_t^l$. The correlation response label of the filter is expressed by $\mathbf{y}_t \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}}$. This function minimizes the ℓ_2 error of the correlation response of the desired correlation filter $\mathbf{f}_t^l$:

$$\varepsilon(\mathbf{f}) = \left\| \sum_{l=1}^{D} \mathbf{x}_t^l \circ \mathbf{f}_t^l - \mathbf{y}_t \right\|_2^2 + \lambda \sum_{l=1}^{D} \left\| \mathbf{f}_t^l \right\|^2, \tag{1}$$

where $\circ$ denotes a circular convolution, and λ is a regularization weight. The response label $\mathbf{y}$ is usually expressed by a Gaussian function [10].

Because Equation (1) is a linear least squares problem, it can be computed efficiently in the Fourier domain by Parseval's formula. Hence, the filter that minimizes Equation (1) is expressed as:

$$\varepsilon(\mathbf{F}) = \left\| \sum_{l=1}^{D} \overline{\mathbf{X}}^l \odot \mathbf{F}^l - \mathbf{Y} \right\|_2^2 + \lambda \sum_{l=1}^{D} \left\| \mathbf{F}^l \right\|^2 \tag{2}$$

where the capital letters denote the discrete Fourier transform (DFT) of the corresponding quantities, the bar $\overline{\bullet}$ denotes a complex conjugation, and $\odot$ denotes a Hadamard product. Therefore, in the first frame, Equation (2) can be solved as:

$$\mathbf{F}^l = \frac{\overline{\mathbf{Y}} \odot \mathbf{X}^l}{\sum_{k=1}^{D} \overline{\mathbf{X}}^k \odot \mathbf{X}^k + \lambda}, \quad l = 1, \ldots D. \tag{3}$$

The numerator $\mathcal{A}_t^l$ and the denominator $\mathcal{B}_t$ are defined for the t-th frame. An optimal update strategy for the filter $\mathbf{F}_t^l$ in the new sample $\mathbf{x}_t$ is as follows:

$$\mathcal{A}_t^l = (1-\eta)\mathcal{A}_{t-1}^l + \eta \overline{\mathbf{Y}} \odot \mathbf{X}_t^l, \tag{4}$$

$$\mathcal{B}_t = (1-\eta)\mathcal{B}_{t-1} + \eta \sum_{k=1}^{D} \overline{\mathbf{X}}_t^l \odot \mathbf{X}_t^l, \tag{5}$$

where the scalar η is a parameter of the learning rate. The correlation scores $\mathbf{y}_t$ for a new test sample $\mathbf{z}_t$ can be computed in the Fourier domain to detect the change in the position in the new frame t:

$$\mathbf{y}_t = \mathcal{F}^{-1} \left\{ \frac{\sum_{l=1}^{D} \overline{\mathcal{A}}_{t-1}^l \odot \mathbf{Z}^l}{\mathcal{B}_{t-1} + \lambda} \right\}, \tag{6}$$

where $\mathbf{Z}^l$ denotes the l-dimensional features extracted from the frame of pending detection. $\mathcal{F}^{-1}$ is the IFFT. Equation (6) can be used to find the maximum correlation score to determine the position of the current target.

The image feature pyramid of the current sample is constructed in a rectangular area behind the learned translation filter to estimate the scale. The scale of the filter is defined as $S = \left\{ \alpha^n | n = \left\lfloor -\frac{N-1}{2} \right\rfloor, \ldots, \left\lfloor \frac{N-1}{2} \right\rfloor \right\}$. The current target region with a size of W × H is reconstructed to form a series of scale patches I_n of size α^nW × α^nH based on N scale levels. The scale filter has a one-dimensional Gaussian score y_s. The max value $S_t(n)$ of the training sample $x_{t,scale}$ for I_n is the current scale.

Because the computational cost of the fast DSST depends primarily on the FFT, we use PCA for dimensionality reduction to improve the computing speed. To update the target template $\mu_t = (1-\eta)\mu_{t-1} + \eta \mathbf{x}_t$, a projection matrix $\mathbf{P}_t \in \mathbb{R}^{d \times D}$ is constructed for μ_t, where d is the dimension of the compression feature. The current test sample $\mathbf{z}_t$ can be obtained by Equation (7) via the compressed training sample $\mathcal{X}_t = \mathcal{F}\{\mathbf{P}_{t-1}\mathbf{x}_t\}$:

$$\mathbf{y}_t = \mathcal{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{\mathcal{A}}_{t-1}^l \odot \mathcal{Z}_t^l}{\mathcal{B}_{t-1} + \lambda} \right\}, \tag{7}$$

where $\mathcal{Z}_t = \mathcal{F}\{\mathbf{P}_{t-1}\mathbf{z}_t\}$ is the new compressed sample, and $\overline{\mathcal{A}}_{t-1}^l$ and $\mathcal{B}_{t-1}$ are the updated numerator and denominator of the template after feature compression, respectively. Note that the projection matrix $\mathbf{P}_t$ is not calculated explicitly but can be obtained quickly by QR decomposition.

3.2. Long-Term Memory Module

In practical applications, trackers often operate for a long time. Objects outside the field of view and occlusions are the main problems in long-term tracking because a target typically does not reappear in the same position where it disappeared. Coal mines have few and uneven light sources, resulting in low illumination. When the scene changes from

a dark to a light environment, the camera is suddenly exposed to the light source, which represents a crucial problem that has to be solved in long-term tracking. Therefore, the tracker needs to detect the location of the target in the new scene and relate it to the sample model in the previous scene to ensure consistent tracking.

The Long-Term Memory Module has two parts: the long-term filter $\mathbf{f}_{\text{Long}}$ and the long-time memory template $\mathbf{x}_{t,\text{Long}}$. Inspired by the LCT [18], we use a long-term filter $\mathbf{f}_{\text{Long}}$ to prevent a tracking failure caused by noise interference during long-term tracking. Unlike in the LCT, we do not use the kernel trick [20] for $\mathbf{f}_{\text{Long}}$ to calculate the response score. Instead, we use a DSST-like approach to calculate the correlation response between the $\mathbf{x}_{t,\text{Long}}$ and $\mathbf{f}_{\text{Long}}$ directly. Note that the $\mathbf{x}_{t,\text{Long}}$ of the $\mathbf{f}_{\text{Long}}$ can also use projection matrix $\mathbf{P}_t$ for feature compression. The sample obtained from the translation filter is used as the detection sample of the long-term filter. The confidence score of each tracked target $\mathcal{Z}_t = \mathcal{F}\{\mathbf{P}_{t-1}\mathbf{z}_t\}$ is computed as $C_{t,\text{Long}} = \max(\mathbf{f}_{\text{Long}}(\mathcal{Z}_t))$. Moreover, the long term memory template $\mathbf{x}_{t,\text{Long}}$ is updated to the current detection sample when the confidence score is above the predefined threshold.

3.3. Re-Detection Module

The re-detection module is a crucial component to improve the robustness and long-term tracking ability of the tracker. It is used to find the target quickly after it has been lost. Our method is based on the LCT+ [30] method, and an online support vector machine (SVM) classifier is used as the detector. The difference is that another confidence parameter (Section 3.4) is used as the criterion in conjunction with the predefined re-detection threshold T_r. The re-detector only trains the translated samples to reduce the computational burden. The feature representation of the sample is based on the multiple experts using the entropy minimization (MEEM) method [40], namely the quantized color histogram. For a training set $\{(\mathbf{v}_i, c_i)i = 1, 2, \ldots, N\}$ with N samples in a frame, the objective function of solving the SVM detector hyperplane $\mathbf{h}$ is:

$$\begin{aligned} &\min_h \tfrac{\lambda}{2}\|\mathbf{h}\|^2 + \tfrac{1}{N}\textstyle\sum_i \ell(\mathbf{h}; (\mathbf{v}_i, c_i)), \\ &\textit{where } \ell(\mathbf{h}; (\mathbf{v}_i, c_i)) = \max\{0, 1 - c\langle \mathbf{h}, \mathbf{v}\rangle\}, \end{aligned} \tag{8}$$

where $\mathbf{v}_i$ represents the feature vector generated by the i-th sample, and $c_i \in \{+1, -1\}$ represents the class label. The notation $\langle \mathbf{h}, \mathbf{v}\rangle$ represents the inner product of the vectors $\mathbf{h}$ and $\mathbf{v}$. A passive-aggressive algorithm is used to update the hyperplane parameters:

$$\mathbf{h} \leftarrow \mathbf{h} - \frac{\ell(\mathbf{h}; (\mathbf{v}, c))}{\left\|\nabla_{\mathbf{h}}\ell(\mathbf{h}; (\mathbf{v}, c))\right\|^2 + \frac{1}{2\tau}} \nabla_{\mathbf{h}}\ell(\mathbf{h}; (\mathbf{v}, c)), \tag{9}$$

where the gradient of the loss function $\mathbf{h}$ is denoted by $\nabla_{\mathbf{h}}\ell(\mathbf{h}; (\mathbf{v}, c))$, and $\tau \in (0, +\infty)$ is a hyper-parameter used to control the $\mathbf{h}$ update rate. Similar to the long-term filter $\mathbf{f}_{\text{Long}}$, we use Equation (9) to update the classifier parameters only when $C_{t,\text{Long}} \geq T_a$.

3.4. Confidence Function and Update Strategy

The tracking confidence parameter is an index for evaluating if the target has been lost. Most DCF-based trackers use the maximum response $R_{\max}$ to locate the target in the next frame. However, in a complex scene, it is not ideal to rely only on this parameter. Wang et al. proposed large margin object tracking method with circulant feature maps (LMCF) [41] with average peak-to-correlation energy (APCE) (Equation (10)), which can effectively deal with the target occlusion and loss. Zhang et al. proposed a motion-aware correlation filter (MACF) [42] based on the confidence of squared response map (CSRM) (Equation (11)), which compensated for the lack of APCE discrimination during long-term occlusion:

$$APCE = \frac{|R_{\max} - R_{\min}|^2}{mean\left(\sum_{w,h}(R_{w,h} - R_{\min})^2\right)}, \tag{10}$$

$$CSRM = \frac{\left|R_{\max}^2 - R_{\min}^2\right|^2}{mean\left(\sum_{w,h}\left(R_{w,h}^2 - R_{\min}^2\right)^2\right)}, \tag{11}$$

where R_{max}, R_{min}, and $R_{w,h}$, respectively, denote the maximum, minimum, and the w-th row h-th column elements of the peak value of the response. A comparison was conducted to determine whether the combination of multiple confidence parameters improved the tracker's performance. The results are presented in Section 4.

In the traditional DCF-based tracker [11–13], it is common to train a sample and the filter in each frame and update the filter. Although an iterative search can be conducted effectively, updating the filter in each frame increases the computational complexity because the optimization of the filter is the core calculation step in the algorithm. We adopt the ECO [14] method to reduce computational complexity by updating the filter template in every N_S-th frame. This strategy improves the running speed of the filter and prevents overfitting. However, the target sample is updated in each frame.

3.5. Dealing with Low-Illumination Environments

3.5.1. Low-Light Image Enhancement

Image enhancement algorithms have been presented by several researchers [43–45]. This study uses the low illumination image enhancement (LIME) method presented by Guo et al. [43]. The model of low illumination image has the following forms:

$$1 - \mathbf{L} = (1 - \mathbf{R}) \odot \widetilde{\mathbf{T}} + \delta(1 - \widetilde{\mathbf{T}}), \tag{12}$$

where $\mathbf{L}$ is the captured image (low illumination), $\mathbf{R}$ is the desired recovery, $\mathbf{T}$ represents the illumination map, and δ is the global atmospheric light (global illumination). This model (Equation (12)) [46] is based on inverted low-light images $1 - \mathbf{L}$, which look like hazy images. LIME first estimates an initial illumination map and then refines it in the second step. In the first step, the following preliminary initial estimates of the non-uniform lighting for each individual pixel p are used:

$$\hat{\mathbf{T}}(p) \leftarrow \max_{c \in \{R,G,B\}} \mathbf{L}^c(p), \tag{13}$$

The goal is to ensure that the obtained $\hat{\mathbf{T}}(p)$ recovery is not saturated; c is the maximum value of the three color channels; a small constant ε is defined to avoid a zero denominator:

$$\mathbf{R}(p) = \frac{\mathbf{L}(p)}{\left(\max_c \mathbf{L}^c(p) + \varepsilon\right)} \tag{14}$$

Subsequently, the atmospheric light δ is substituted into the $1 - \mathbf{L}$ model Equation (12):

$$\widetilde{\mathbf{T}}(p) \leftarrow 1 - \delta^{-1} + \max_c \mathbf{L}^c(p) \cdot \delta^{-1}, \tag{15}$$

$$\mathbf{R}(p) = \frac{\mathbf{L}(p) - 1 + \delta}{\left(1 - \delta^{-1} + \max_c \mathbf{L}^c(p) \cdot \delta^{-1} + \varepsilon\right)} + (1 - \delta), \tag{16}$$

The initial illumination map $\hat{\mathbf{T}}(p)$ is obtained by Equation (13) due to its conciseness, and the refined illumination map $\mathbf{T}(p)$ is obtained using the following optimization function:

$$\min_{\mathbf{T}(p)} \left\| \hat{\mathbf{T}}(p) - \mathbf{T}(p) \right\|_F^2 + \beta \|\mathbf{W} \odot \nabla \mathbf{T}(p)\|_1, \tag{17}$$

where β is a regularization weight, $\|\bullet\|_F$ and $\|\bullet\|_1$ designate the Frobenius and ℓ_1 norms, respectively. $\mathbf{W}$ is a weight matrix, and ∇ is the first-order derivative filter consisting of $\nabla_h \mathbf{T}(p)$ (horizontal) and $\nabla_v \mathbf{T}(p)$ (vertical). The relative total variation (RTV) [47] and

the two-throughout Gaussian kernel Equation (18) were used as the standard deviation σ to select the weight matrix $\mathbf{W}$. For each location, the weight $\mathbf{W}_o(p)$ is determined in the following manner; the subscript o represents the orientation of the element (h or v):

$$G_\sigma(p,q) \propto \exp\left(-\frac{dist(p,q)}{2\sigma^2}\right), \tag{18}$$

$$\mathbf{W}_o(p) \leftarrow \sum_{q\in\Omega(p)} \frac{G_\sigma(p,q)}{\left|\sum_{q\in\Omega(p)} G_\sigma(p,q)\nabla_o\hat{\mathbf{T}}(q)\right| + \varepsilon}, \tag{19}$$

where $\Omega(p)$ is a region centered at pixel p, q is the location index within the region, and $|\bullet|$ is the absolute value operator. Equation (17) can be approximately calculated by the following:

$$\min_{\mathbf{T}} \left\|\hat{\mathbf{T}}(p) - \mathbf{T}(p)\right\|_F^2 + \beta\sum_p \frac{\mathbf{W}_h(p)(\nabla_h\mathbf{T}(p))^2}{\left|\nabla_h\hat{\mathbf{T}}(p)\right| + \varepsilon} + \frac{\mathbf{W}_v(p)(\nabla_v\mathbf{T}(p))^2}{\left|\nabla_v\hat{\mathbf{T}}(p)\right| + \varepsilon}, \tag{20}$$

Hence, Equation (13) can be employed to initially estimate the illumination map $\hat{\mathbf{T}}(p)$. After the refined illumination map $\mathbf{T}(p)$ is obtained by Equation (17), $\mathbf{R}$ can be recovered by Equation (14).

3.5.2. Accelerated Versions for Mobile Drilling Robot

The LIME method is employed for enhancing a single image, but the method does not consider the real-time performance using video data. Although LIME can preserve image features well and has high tracking accuracy (Section 4.4), its slow running speed makes it unsuitable for mobile robot applications. Therefore, a fast image enhancement method is proposed to ensure that the LLCT can be used on mobile robots. The brightness value is extracted from the three image channels (RGB):

$$Light = \frac{1}{2}\left(\max_{c\in\{R,G,B\}} \mathbf{L}^c(p) + \min_{c\in\{R,G,B\}} \mathbf{L}^c(p)\right), \tag{21}$$

The average values of the RGB channels are calculated, and the maximum and minimum mean values are used as the brightness values. An effective nonlinear superposition algorithm is iterated to improve the image brightness:

$$\mathbf{R}^c(p) = \mathbf{L}^c(p) + k\mathbf{L}^c(p) \odot \left(\frac{255 - \mathbf{L}^c(p)}{256}\right) \quad c \in \{R,G,B\}, \tag{22}$$

where $\mathbf{L}(\bullet)$ denotes the captured image, $\mathbf{R}(\bullet)$ denotes the desired image, and $k \in [0,1]$ is a parameter for controlling the exposure. Equation (22) is iterated until the brightness reaches the threshold value. The target feature is then extracted for filter training.

Because the global brightness does not reflect whether the target is in a low-illumination environment (Figure 3), we propose a target area illumination detection method. We only consider the illuminance of BB and its surrounding padding area. When the brightness is below the predetermined threshold T_L, the exposure of the current frame is adjusted. This step is skipped when the illumination is sufficient.

3.6. Visual Tracking System Framework for Drilling Robot

The hardware and software framework of the visual tracking system for the coal mine drilling robot is shown in Figure 4. In this system, the sensor layer consists of monocular color cameras, 3D Lidar, and other sensors. We used only the camera for the acquisition of raw images in this work. The captured images can be manually framed for target selection via the operator interface or automatically framed for target selection using the target recognition algorithm. After confirming the target to be tracked, the LLCT module

performs the target tracking and feeds the position back to the on-board computer and controller. This is used for subsequent robot formation tracking and other functions.

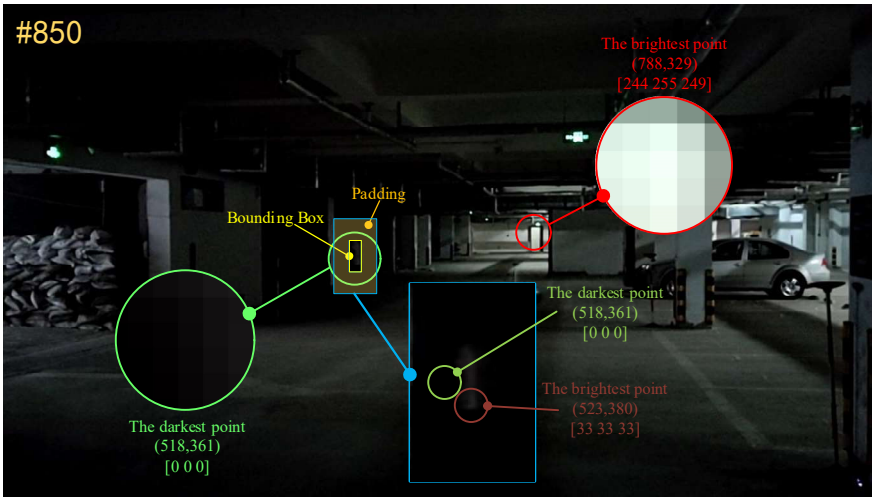

Figure 3. Selection of the low-illumination detection area. In the full view (1280 × 720), the darkest pixel (green circle) is close to the target, but the brightest (red circle) is not. The maximum RGB values close to the target are [33 33 33]. The global brightness of the image is 128, but the brightness close to the target is only 16.5.

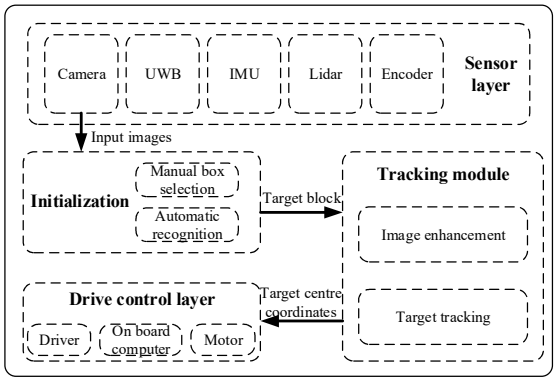

Figure 4. Hardware and software framework of vision tracking system for coal mine drilling robot.

The software system uses ROS [48] for interaction, i.e., a communication node approach. The sensor drive module, the target tracking module, the SLAM module, and the path planning modules operate independently in the lower computer system. The host system runs the visualization interface and the remote control command-sending module. Each system and module interacts by subscribing to the appropriate topic or requesting the appropriate service. The target tracking nodes can, therefore, be switched on independently when there is a demand for them.

4. Experimental

4.1. Implementation Details

The dataset to evaluate the LLCT was the Online Tracking Benchmark (OTB) [1], an authoritative dataset that has been used to compare the performance of other trackers (Section 4.3.1). It contains 50 sequences and has many challenging attributes. Moreover, we also evaluated the LLCT performance on the classic evaluation datasets and benchmarks UAV123 [4] (Section 4.3.2). The proposed tracker was implemented in MATLAB 2016b on an industrial computer with Intel i7-8700 3.70 GHz CPU and Nvidia GeForce GTX 1080 GPU. The CNN feature was not used to improve the real-time performance of the

operation, and only manually selected features (histogram of oriented gradients (HOG) and color names) were used. Therefore, the separate graphics card was disabled to reduce power consumption.

The regularization parameter λ was 0.01, and the learning rate η was 0.025. The standard deviation of the Gaussian function output y was 1/16 of the translation target size. The padding of the filter was twice the size of the initial target. The scale filters were interpolated from N = 17 scales to N^* = 33 scales by interpolation, and the scale factor a = 1.02 was used. The re-detection threshold T_r was 0.2 for the activation detection module and T_a = 0.4 for the detection result. Note that the threshold setting here is only a fraction of the long-term filter $\mathbf{f}_{\text{Long}}$. The confidence parameters of the LCT response were 0.9 times the maximum response and 0.75 times the APCE (or CSRM) response. The long-term memory template was updated when both its parameters exceeded the set threshold. The purpose of this was to test the effect of multi-confidence settings on the trackers' performance. The exposure control parameter k of the LLCT was 1. The lighting intensity threshold T_L = 48. The number of iterations for the image enhancement was 1 (Section 4.4).

The OTB-2013 dataset [1] results were evaluated by the overlap precision (OP), distance precision (DP), and tracking speed (FPS). The success plots show the DP rate [0, 1] in 20 pixels and the area-under-the-curve (AUC) of the OP rate.

4.2. Update Strategy Comparison

The image enhancement module and the tracking module are relatively independent. Therefore, the calculation time for the image enhancement module is calculated independently of the demand, so we tested the interval update strategy without the image enhancement module enabled. The update gap was N_S = 1, 3, 5, and the performance (AUC) was compared using the running speed. The experimental results are presented in Table 1 and Figure 5. It was found that the best update performance was obtained for N_S = 3, but the running speed was significantly higher for N_S = 5. Therefore, the filter was updated every three frames in the subsequent application. The filter does not perform interval updates from the first frame. Experience has shown updating after 10–20 frames substantially improve the tracker's robustness due to less noise interference [14].

Table 1. Update strategy comparison.

N_S	AUC	FPS
1 (per frame)	58.6%	29.5
3	61.3%	32.8
5	60.6%	37.5
f-DSST	60.0%	173

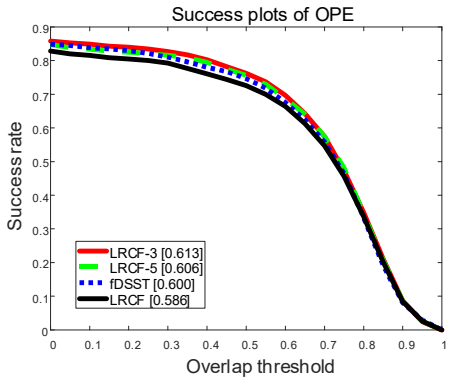

Figure 5. The OTB-2013 benchmark test for the interval update strategy. The effect was obviously improved after adding long-term memory filter. In an occlusion environment, the performance when N_S = 3 was not as good as when N_S = 5, presumably because of the occlusion time, and the occlusion object is trained as a sample.

4.3. Overall Performance on Benchmark Dataset

4.3.1. Performance on OTB-2013

The proposed method was compared with the trackers presented by Wang et al. [1] and the baseline trackers TLD [29], LCT [18], and fast DSST [39]. Figure 6 presents the comparison of our method with the baseline trackers. The proposed method provides superior real-time performance, which an average running speed of 30 FPS. Figure 7 demonstrates the superiority of the proposed method in difficult tracking scenarios, such as low resolution, fast motion, scale variation, and occlusion. Compared with the fast DSST, the proposed method shows improvements in the DP of 13.1%, 3.0%, 2.1%, and 6.6%, respectively. It was found that using multiple confidence parameters in the filter updates degraded the performance.

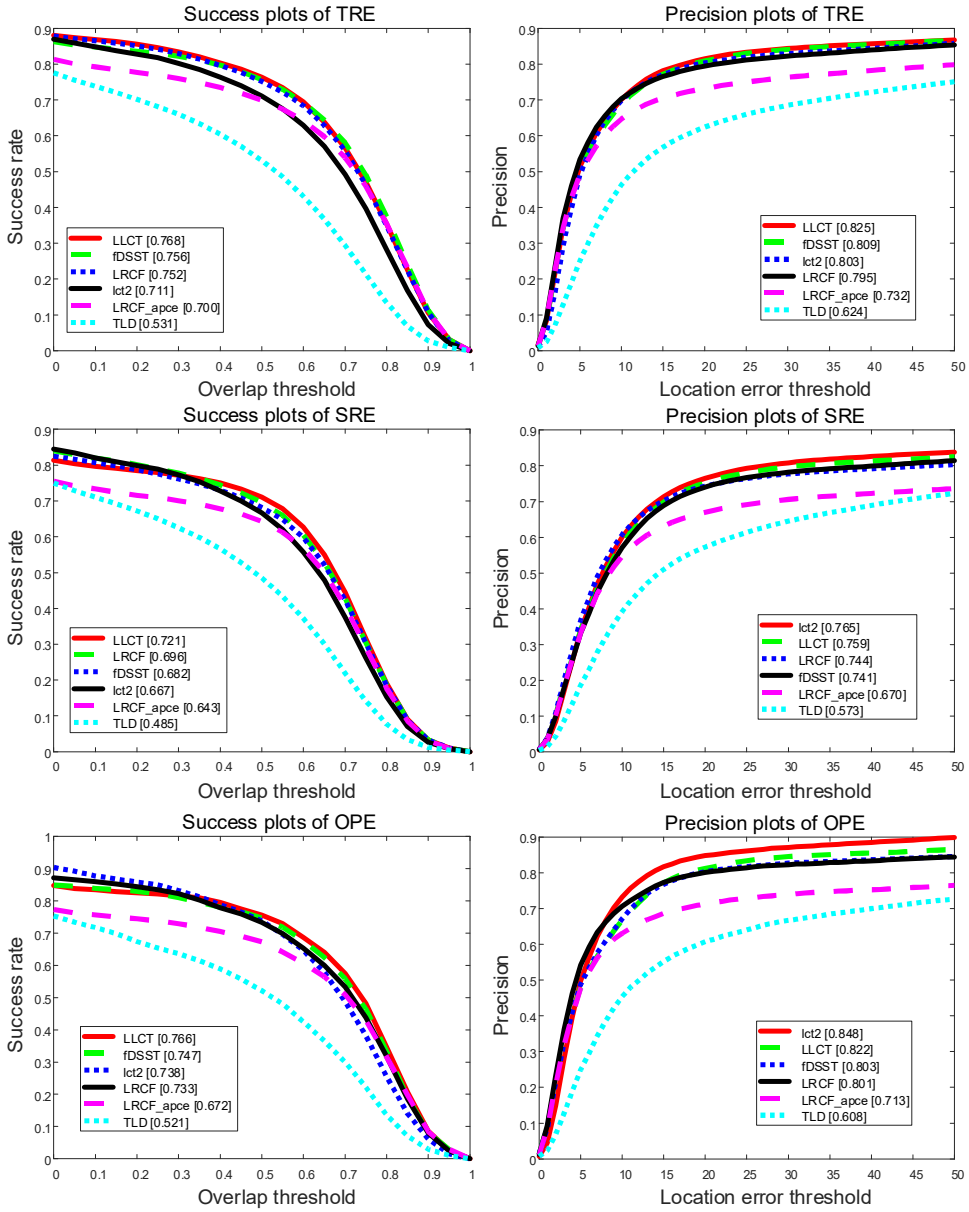

Figure 6. Distance precision and overlap success plots of the average overall performance on the OTB dataset. Temporal robustness evaluation (TRE), spatial robustness evaluation (SRE), and one pass evaluation (OPE) are presented in this figure. For more details, please refer to [1].

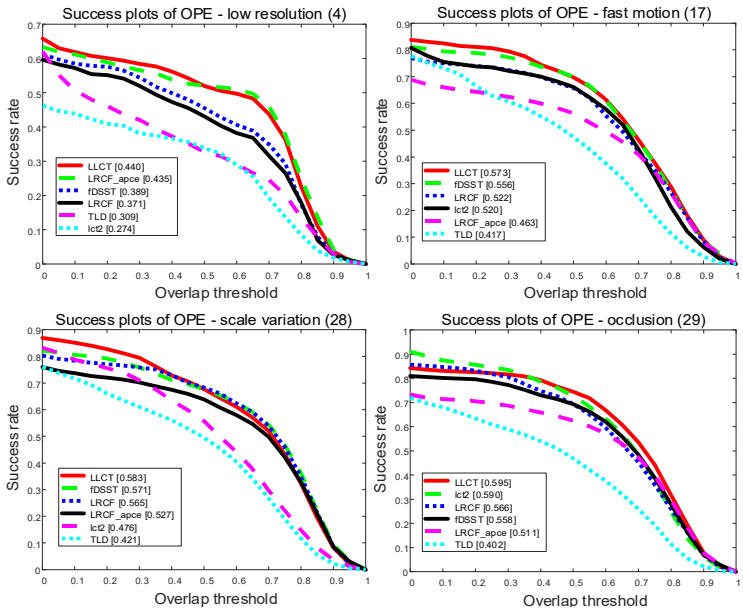

Figure 7. The results of the proposed method and the baseline for different challenging conditions. The AUC score for each tracker is reported in the legend.

4.3.2. Performance on UAV123

The proposed tracker was also evaluated on the UAV123 [4] benchmark dataset, which contains 123 short, challenging videos and 20 long, challenging videos. UAV123 contains twelve challenging attributes, namely (i) Scale Variation, (ii) Aspect Ratio Change, (iii) Low Resolution, (iv) Fast Motion, (v) Full Occlusion, (vi) Partial Occlusion, (vii) Out-of-View, (viii) Background Clutter, (ix) Illumination Variation, (x) Viewpoint Change, (xi) Camera Motion and (xii) Similar Object.

Figure 8 shows the overlap success rate of the proposed method and the baseline trackers. Our proposed tracker achieved the second-highest mean overlap success rate of 0.418 and a mean OP of 0.626. The BACF [12] provides the best performance with a slightly higher success rate, whereas the remaining three trackers, i.e., LRCF [17], fast DSST [39], and LCT [18], have to mean overlap success rates of 0.382, 0.375, and 0.334, respectively. The proposed tracker shows considerable performance on UAV123 (Table 2). The LLCT exceeds the performance of the benchmark trackers (fast DSST and LCT) regarding all challenging attributes, and its performance is almost the same as that of the SOTA tracker in the same period.

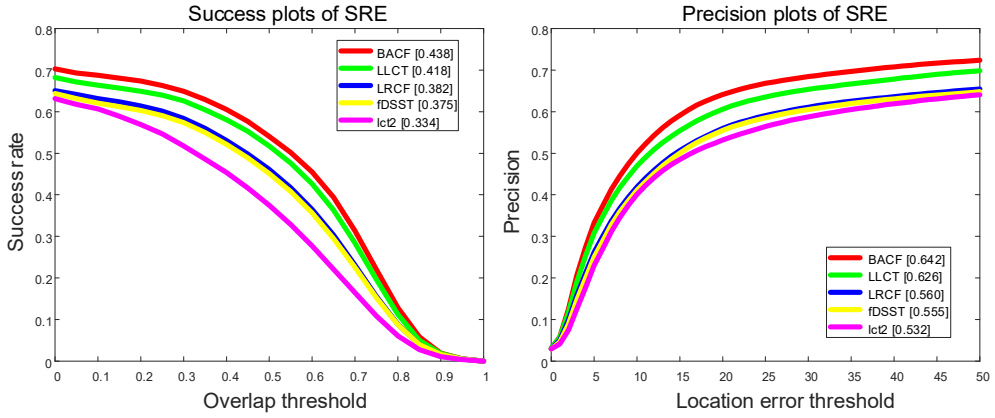

Figure 8. Quantitative results: the overlap success rate and distance precision for a threshold of 20 pixels.

Table 2. The average precision on UAV123.

Attribute	LLCT	LRCF	BACF	f-DSST	LCT
SV	0.555	0.524	0.598	0.517	0.479
ARC	0.510	0.475	0.548	0.477	0.442
LR	0.527	0.488	0.520	0.464	0.420
FM	0.373	0.354	0.493	0.343	0.276
FOC	0.387	0.378	0.425	0.383	0.382
POC	0.530	0.473	0.555	0.466	0.462
OV	0.420	0.407	0.511	0.415	0.398
BC	0.491	0.453	0.513	0.444	0.478
IV	0.495	0.482	0.510	0.482	0.439
VC	0.506	0.486	0.571	0.493	0.444
CM	0.551	0.497	0.618	0.494	0.491
SOB	0.654	0.624	0.668	0.601	0.581

4.4. Experiments in Low-Illumination Environments

We acquired an image sequence in an underground garage to determine the performance of the proposed algorithm under low-illumination conditions. The video was 34 s long and contained 1044 frames. The target was subjected to light changes, out-of-plane rotation, a similar background, and an almost completely dark area starting in the 800th frame.

The effect of illumination discrimination on image enhancement is shown in Figure 9. A bright spot occurs in the lower-left corner of the image in the 83rd frame, resulting in a global brightness value of 120, although the brightness around the target is only 21. The BB discrimination performs well for image enhancement; the global illumination is 120 in the 83rd frame. The standard *David* sequence is calibrated starting in the 300th frame, but the target is already visible in the 150th frame (*Light* = 30). This sequence indicates that the threshold T_L does not need to be very high. However, the disadvantage is observed in the *Garage* sequence (Figure 3). In Figure 6, which depicts the *Garage* sequence after the 800th frame, the global illuminance is at the maximum, but the BB illuminance is less than 20. The target cannot be distinguished from the surrounding environment. The exposure control parameter k was set to 1, and the lighting intensity threshold T_L was conservatively set to 48.

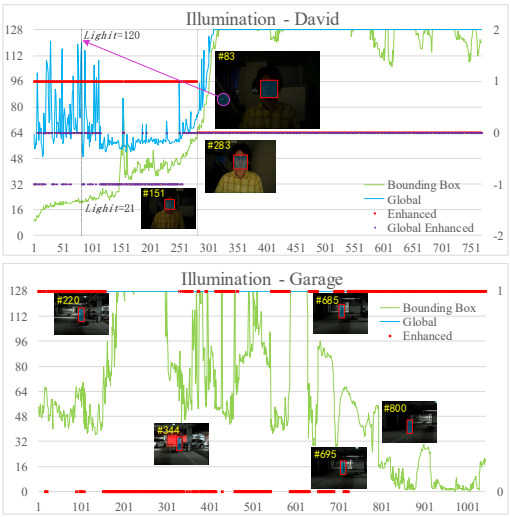

Figure 9. Comparison of global and BB illumination. The top figure shows the luminance values of the *David* sequence, and the following is the *Garage* sequence. The BB and global illumination increase over time in the David sequence. However, in the *Garage* sequence, the global brightness is always the maximum because of the lamp on the right.

Figure 10 presents the comparison of the two modes of LIME, the MATLAB function *imadjust*, and the accelerated version, in terms of the time cost. The first graph in Figure 10 presents the comparison curves of the four solvers (accelerated version, *imadjust*, LIME-exact, and LIME-speed solvers) in terms of time cost. The sample image is scaled to different levels from the original size (3144 × 3078). The results show that the four solvers are sufficiently efficient when the image size is smaller than 400. However, the histogram indicates that the LIME solvers (the exact solver and the speed solver) require several seconds to calculate a frame when the image size is small. In contrast, the proposed method requires only milliseconds to perform an iterative calculation. For a 720P resolution image, the accelerated version is 20 times faster than the accelerated LIME version, and it can run at around 60 FPS. Our method is slightly faster than the MATLAB function. When the image size exceeds 1000, there is a sharp difference in the time cost between the four enhancement methods. The frame rate of our method remains on the order of milliseconds, whereas LIME requires seconds. Thus, the accelerated version of the image enhancement algorithm has a low computational cost, meeting the real-time requirements of mobile robots equipped with 720P resolution cameras.

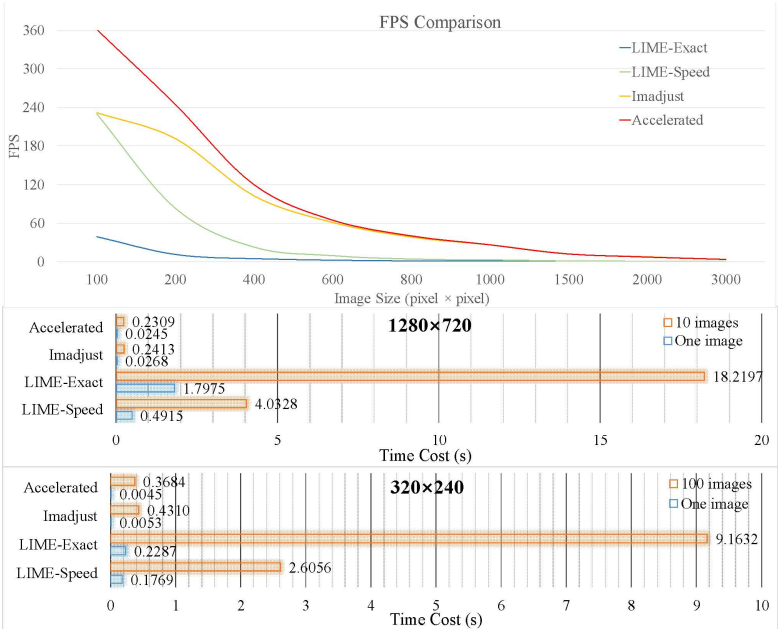

Figure 10. Comparison of the FPS of the algorithms and the cost of several image enhancement methods with different image sizes.

Figure 11 presents the comparison of our tracker (yellow) with the benchmark trackers, e.g., LRCF [17] (blue), fast DSST [39] (green), and LCT [18] (red). The proposed tracker was also compared with the SOTA trackers BACF [12] (orange) and ECO [14] (cyan) at the time and the latest ADT [36] (purple). In the 165th frame (background similarity), the LCT and ECO fail, following the car after the 320th frame. After the 810th frame (dark area), the LRCF, fast DSST, and BACF remain in their original positions, and only LLCT tracks the target to the last frame. Table 3 shows the average operating speed of the trackers and their performance. The LLCT-E and LLCT-S (LIME-Exact and LIME-Speed version) have the lowest processing speed of all trackers for the 720P resolution images, and the LLCT-A (accelerated version) maintains the real-time speed. The LLCT trackers have high accuracies and success rates, and the other baseline trackers except ADT fail to track in a dark environment or with similar backgrounds. Thus, the proposed method outperforms the other SOTA trackers.

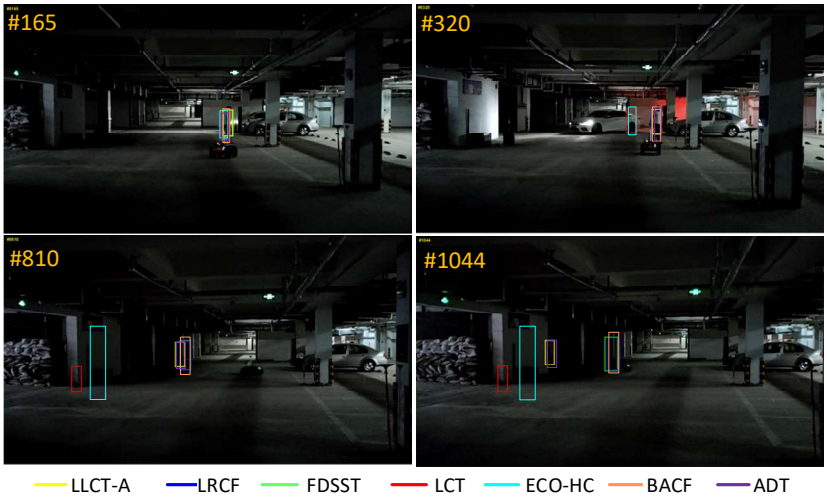

Figure 11. Field experiment in the underground garage. The proposed method performed well in a low-illumination environment.

Table 3. The average operating speed and performance.

Tracker	LLCT-A	LLCT-E	LLCT-S	LRCF	f-DSST	LCT	ECO-HC	BACF	ADT
FPS	38.9	1.00	4.49	35.6	146	48.59	55.69	40.36	38.04
DP	100% [1]	100%	100%	68.8%	32.5%	28.6%	28.4%	67.1%	99.4%
OP	93.2%	95.6%	96.1%	17.1%	27.4%	24.1%	17.7%	46.0%	93.1%

[1] Red: the best; Green: the second; Blue: the third. OP: average overlap score with the threshold of 0.5, DP: average center location error with the threshold of 20 pixels.

Figure 12 shows that the image details obtained from the LIME [43] are excellent, the noise control is effective, and only one overexposed area occurs on the left side of the image. The proposed method does not perform well for image noise reduction, and almost all areas other than the low-illumination area are overexposed. In practical application, LIME better addresses the restoration of image details. Figures 11 and 12 indicate that the accurate enhancement method has almost the same target tracking performance as the "rough" enhancement method when the target is in a dark area. Additionally, the proposed enhancement method has lower computational complexity and reasonable real-time performance of the tracker. This means that the tracker does not require hyperfine samples and can track the target accurately according to the difference between the target and the background.

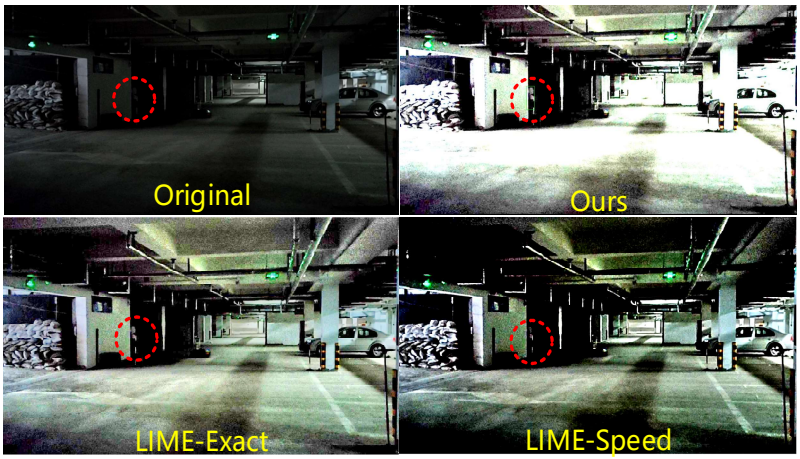

Figure 12. This is a figure. Schemes follow the same formatting.

4.5. Field Experiment in A Coal Mine

We conducted a field experiment in a coal mine in Pingdingshan, China, to evaluate the proposed tracker. The experimental platform was the ZDY4000LK coal mine drilling robot developed by the China Coal Technology Engineering Group (CCTEG) at the Xi'an Research Institute. The experimental environment and hardware configuration are shown in Figure 13. The experiment was conducted at the air shaft of a ventilation roadway in a coal mine, where the lighting conditions were poor. An intrinsically safe (Ex i) three-dimensional lidar sensor, an Ex i monocular camera, and two flameproof (Ex d) monocular cameras were installed at the front of the robot. Two of the cameras faced the wall, and one was pointed away from the robot. The forward-facing camera was used for real-time image acquisition, and the data were stored in the rosbag format. Lidar collects 3D point cloud to provide navigation information for a robot. The image resolution was 1280 × 720 pixels, and 940 images were acquired. The video data contained challenges such as light changes, out-of-plane rotation, out-of-view, and fast movement.

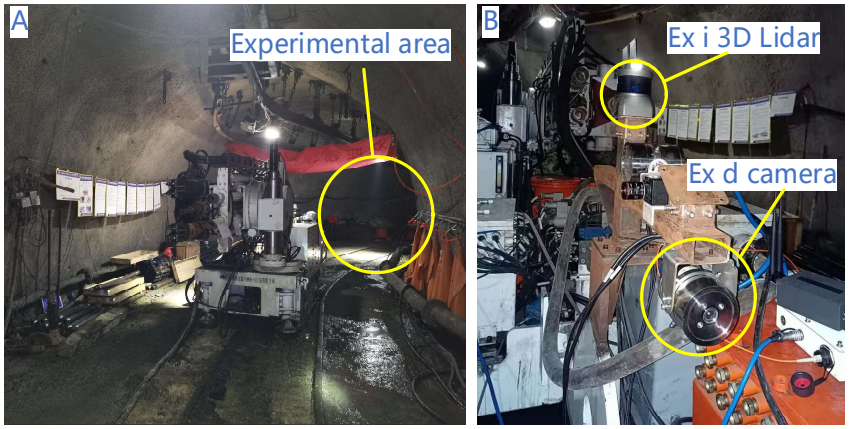

Figure 13. Field experiment in a coal mine roadway. (**A**) shows the roadway environment and the location of the drilling robot. (**B**) shows the location of the external sensors, which meet the underground explosion-proof requirements.

In the field experiment, we use the C++ version of the algorithm and it runs in the ROS. The target area is manually selected in the camera interface. The parameters are slightly adjusted. We changed the search area shape as "square" instead of "proportional", and the search area scale was set as 5. The CSRM confidence was enabled. Other parameters are consistent with the dataset evaluation.

Figure 14 shows the tracking results of the LLCT, baseline trackers (LCT [18], fast DSST [39]), and the SOTA (BACF [12], ECO-HC [14], ADT [36]) for tracking a dynamic object in the downhole environment. The proposed tracker outperforms most of the SOTA trackers under these conditions. In the first 450 frames, all algorithms can track the moving target accurately, but some trackers show offsets (e.g., LCT and fast DSST). When the target rotates out of the plane around the 500th frame, the LCT and fast DSST lose the target, and a target offset occurs in the BACF. When the target moves out of view in the 600th frame, the LLCT, BACF, and ECO-HC remain at the position where the target disappears. After the target reappears in frame 635, ADT and LLCT reacquire the target. Overall, LLCT performed better than ADT. Table 4 shows the quantitative results, which indicate that the tracking success rate of the LLCT is higher than that of the other trackers.

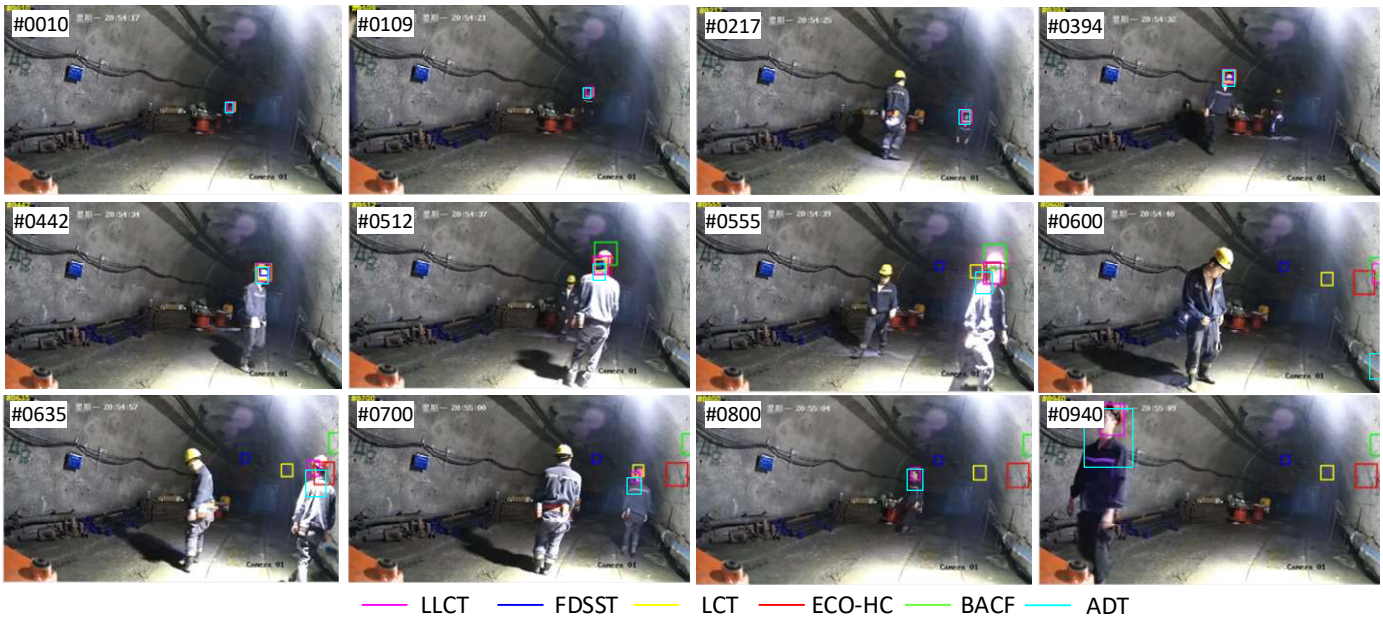

Figure 14. The qualitative experiment comparing the LLCT with state-of-the-art trackers in the coal mine roadway. The proposed method performed well in the coal mine environment.

Table 4. The quantitative performance of the field experiment.

Tracker	LLCT	f-DSST	LCT	ECO-HC	BACF	ADT
FPS	36.6	154 [1]	51.8	56.7	52.4	35.6
DP	80.8%	8.72%	55.9%	57.1%	47.7%	48.2%
OP	65.3%	7.34%	48.2%	43.9%	38.1%	52.8%

[1] Red: the best; Green: the second; Blue: the third. OP: average overlap score with the threshold of 0.5, DP: average center location error with the threshold of 20 pixels.

Figure 15 shows the comparison of the predicted trajectory obtained from the LLCT tracker and the actual trajectory. The trajectory tracking strategy refers to MACF [42]. In the experiments, the depth information is replaced by the scale information of each frame, and the initial scale is set to $S_{\text{init}}(n) = 1.0$, i.e., $n = 0$. A larger $S_t(n)$ value means that the target is closer to the robot. The blue line represents the track fitted by the benchmark center point, the yellow line is the track fitted by the LLCT at the center of the tracking BB, and the red line is the filtered track obtained from a Kalman filter. The results in the 2D plane and 3D space indicate that the LLCT tracker accurately predicts the position and scale of moving targets in a static background.

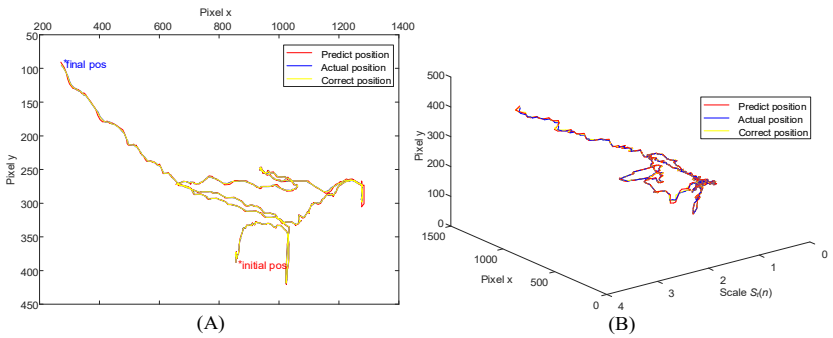

Figure 15. The position trajectory of the object. (**A**) The predicted, actual, and corrected positions in the plane. (**B**) The predicted, actual, and corrected positions in the 3D space; the depth is approximately replaced by the target scale.

5. Conclusions

To address the impact of low illumination environments on vision tracking algorithms for coal mine drilling robots. An effective tracker for mobile robot applications, LLCT, was proposed for long-term VOT in low-illumination environments. The trick of fast DSST was used to calculate the image correlation instead of the kernel convolution in LCT, and a projection matrix was incorporated into the traditional DCF filter to reduce the dimensionality of the extracted sample features. An effective method was proposed for the detection of target illumination, which allows an accurate estimation of the brightness around the target. An image enhancement module was added to achieve tracking in low-illumination environments and proposed a fast image enhancement method, which can run at a frame rate of around 60 FPS at 720P resolution. The experimental results demonstrated that the LLCT had good robustness and excellent performance on the OTB-2013 benchmark and UAV123 benchmark. Finally, a low-light vision tracking system based on the ROS operating system was designed and successfully applied to a coal mine drilling robot. In low illumination image sequences, the proposed tracker improves performance by more than 200% over the baseline tracker and by more than 50% over its contemporaries. A field experiment was conducted with an underground drilling robot in a coal mine. The results revealed that the proposed LLCT was superior to other methods for target tracking in low-illumination environments, and it can track the target trajectory correctly.

In the future, we aim to develop an autonomous detection and tracking system based on our current work to achieve visual auto-following between the drilling robot and the drill pipe transporting robot. In addition, the processing speed for large images requires optimization, and the running speed of the image enhancement module must be improved.

Author Contributions: Methodology, S.Y.; software, M.L.; validation, S.Y., M.L. and C.T.; formal analysis, Y.L.; writing—original draft preparation, S.Y.; writing—review and editing, S.Y.; project administration and funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key Research and Development Program of China under Grant (No. 2018YFC0808000); in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions of China (PAPD); in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No.19KJB460014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, Y.; Lim, J.; Yang, M.-H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–27 June 2013; pp. 2411–2418.
2. Kristan, M.; Matas, J.; Leonardis, A.; Vojíř, T.; Pflugfelder, R.; Fernandez, G.; Nebehay, G.; Porikli, F.; Čehovin, L. A novel performance evaluation methodology for single-target trackers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2137–2155. [CrossRef] [PubMed]
3. Liang, P.; Blasch, E.; Ling, H. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [CrossRef]
4. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 10–16 October 2016; pp. 445–461.
5. Smeulders, A.W.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual tracking: An experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1442–1468. [CrossRef]
6. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *Acm Comput. Surv. CSUR* **2006**, *38*, 1–45. [CrossRef]
7. Li, X.; Hu, W.; Shen, C.; Zhang, Z.; Dick, A.; Van Den Hengel, A. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol.* **2013**, *4*, 1–48. [CrossRef]
8. Xu, K.; Chia, K.W.; Cheok, A.D. Real-time camera tracking for marker-less and unprepared augmented reality environments. *Image Vis. Comput.* **2008**, *26*, 673–689. [CrossRef]

9. Wu, Y.; Lim, J.; Yang, M.-H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
10. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
11. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 702–715.
12. Kiani Galoogahi, H.; Fagg, A.; Lucey, S. Learning background-aware correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1135–1143.
13. Lukezic, A.; Vojir, T.; Zajc, L.C.; Matas, J.; Kristan, M. Discriminative Correlation Filter with Channel and Spatial Reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4847–4856.
14. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
15. Zhang, M.; Liu, X.; Xu, D.; Cao, Z.; Yu, J. Vision-Based Target-Following Guider for Mobile Robot. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9360–9371. [CrossRef]
16. Wang, N.; Shi, J.; Yeung, D.-Y.; Jia, J. Understanding and diagnosing visual tracking systems. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3101–3109.
17. You, S.; Zhu, H.; Li, M.; Wang, L.; Tang, C. Long-Term Real-Time Correlation Filter Tracker for Mobile Robot. In Proceedings of the International Conference on Intelligent Robotics and Applications, Shenyang, China, 8–11 August 2019; pp. 245–255.
18. Ma, C.; Yang, X.; Zhang, C.; Yang, M.H. Long-term correlation tracking. In Proceedings of the Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5388–5396.
19. Danelljan, M.; Khan, F.S.; Felsberg, M.; Weijer, J.V.D. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097.
20. Henriques, J.F.; Rui, C.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef] [PubMed]
21. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 10–16 October 2016; pp. 850–865.
22. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 10–16 October 2016; pp. 472–488.
23. Ma, C.; Huang, J.B.; Yang, X.; Yang, M.H. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
24. Zhang, Z.; Xie, Y.; Xing, F.; Mcgough, M.; Lin, Y. MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3549–3557.
25. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.
26. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.155. [CrossRef]
27. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Convolutional Features for Correlation Filter Based Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshop, Santiago, Chile, 7–13 December 2015; pp. 621–629.
28. Yuan, D.; Li, X.; He, Z.; Liu, Q.; Lu, S. Visual object tracking with adaptive structural convolutional network. *Knowl.-Based Syst.* **2020**, *194*, 1–11. [CrossRef]
29. Zdenek, K.; Krystian, M.; Jiri, M. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. [CrossRef]
30. Ma, C.; Huang, J.-B.; Yang, X.; Yang, M.-H. Adaptive correlation filters with long-term and short-term memory for object tracking. *Int. J. Comput. Vis.* **2018**, *126*, 771–796. [CrossRef]
31. Zhu, G.; Wang, J.; Wu, Y.; Lu, H. Collaborative Correlation Tracking. In Proceedings of the British Machine Vision Conference, Swansea, UK, 7–10 September 2015; pp. 1–12.
32. Penrose, R. On best approximate solutions of linear matrix equations. In Proceedings of the Mathematical Proceedings of the Cambridge Philosophical Society, Oxford, UK, 1 January 1956; pp. 17–19.
33. Yan, B.; Zhao, H.; Wang, D.; Lu, H.; Yang, X. 'Skimming-Perusal' Tracking: A Framework for Real-Time and Robust Long-Term Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2385–2393.
34. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4277–4286.

35. Shang, W.; Cao, X.; Ma, H.; Zang, H.; Wei, P. Kinect-based vision system of mine rescue robot for low illuminous environment. *J. Sens.* **2016**, *2016*, 8252015. [CrossRef]
36. Li, B.; Fu, C.; Ding, F.; Ye, J.; Lin, F. ADTrack: Target-aware dual filter learning for real-time anti-dark UAV tracking. In Proceedings of the IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021; pp. 496–502.
37. Ye, J.; Fu, C.; Zheng, G.; Cao, Z.; Li, B. DarkLighter: Light Up the Darkness for UAV Tracking. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3079–3085.
38. Ye, J.; Fu, C.; Cao, Z.; An, S.; Zheng, G.; Li, B. Tracker Meets Night: A Transformer Enhancer for UAV Tracking. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3866–3873. [CrossRef]
39. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [CrossRef] [PubMed]
40. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the European Conference on Computer Vision, Zurich, The Switzerland, 5–12 September 2014; pp. 188–203.
41. Wang, M.; Liu, Y.; Huang, Z. Large margin object tracking with circulant feature maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4800–4808.
42. Zhang, Y.; Yang, Y.; Zhou, W.; Shi, L.; Li, D. Motion-Aware Correlation Filters for Online Visual Tracking. *Sensors* **2018**, *18*, 3937. [CrossRef] [PubMed]
43. Guo, X. LIME: A method for low-light image enhancement. In Proceedings of the 24th ACM International conferenCe on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 87–91.
44. Dong, X.; Wang, G.; Pang, Y.; Li, W.; Wen, J.; Meng, W.; Lu, Y. Fast efficient algorithm for enhancement of low lighting video. In Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, Barcelona, Spain, 11–15 July 2011; pp. 1–6.
45. Hessel, C. An Implementation of the Exposure Fusion Algorithm. *Image Process. Line* **2018**, *8*, 369–387. [CrossRef]
46. He, K.; Sun, J.; Tang, X. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 2341–2353. [CrossRef] [PubMed]
47. Li, X.; Yan, Q.; Yang, X.; Jia, J. Structure Extraction from Texture via Relative Total Variation. *Acm Trans. Graph.* **2012**, *31*, 1–10. [CrossRef]
48. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; pp. 1–6.