actuators

MDPI

Article

An Intelligent Tracking System for Moving Objects in Dynamic Environments

Nada Ali Hakami [1], Hanan Ahmed Hosni Mahmoud [2] and Abeer Abdulaziz AlArfaj [2,*]

[1] Department of Computer Science, College of Computer Science and Information Technology, Jazan University, Jazan 45142, Saudi Arabia
[2] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
* Correspondence: aaalarfaj@pnu.edu.sa

Abstract: Localization of suspicious moving objects in dynamic environments requires high accuracy mapping. A deep learning model is proposed to track crossing moving objects in the opposite direction. Moving objects locus measurements are computed from the space included in the boundaries of the images in the intersecting cameras. Object appearance is designated by the color and textural histograms in the intersecting camera views. The incorrect mapping of moving objects in a dynamic environment through synchronized localization can be considerably increased in complex areas. This is done due to the presence of unfit points that are triggered by moving targets. To face this problem, a robust model using the dynamic province rejection technique (DPR) is presented. We are proposing a novel model that incorporates a combination of the deep learning method and a tracking system that rejects dynamic areas which are not within the environment boundary of interest. The technique detects the dynamic points from sequential video images and partitions the current video image into super blocks and tags the border differences. In the last stage, dynamic areas are computed from dynamic points and superblock boundaries. Static regions are utilized to compute the positions to enhance the path computation precision of the model. Simulation results show that the introduced model has better performance than the state-of-the-art similar models in both the VID and MOVSD4 datasets and is higher than the state-of-the-art tracking systems with better speed performance. The experiments prove that the computed path error in the dynamic setting can be decreased by 81%.

Keywords: neural network architecture; moving object; localization; tracking system

Citation: Ali Hakami, N.; Hosni Mahmoud, H.A.; AlArfaj, A.A. An Intelligent Tracking System for Moving Objects in Dynamic Environments. *Actuators* **2022**, *11*, 274. https://doi.org/10.3390/act11100274

Academic Editors: Jing Wang, Zhijie Xu, Zhenyu Lu and Jonathan Gomez

Received: 10 August 2022
Accepted: 22 September 2022
Published: 25 September 2022

1. Introduction

The speedy progress of engineering automation can benefit many systems, such as automated environment systems (ABS) which are utilized extensively in robot vision systems, country border, and other dynamic dangerous areas inspections. The established regulated technologies comprise laser and magnetic guidance [1–4]. These guidance processes were used in the past, but their deficiencies are noticeable. Path computation recognizes moving objects in the surveillance areas and utilizes sensors in the automated environment systems to detect and trail these provisions. This technique has poor ability, low accuracy, and anti-interference disadvantage. Laser guidance is characterized by high accuracy but demands an ample reflector to be placed in the surveillance areas to cover all the required areas and overcome the occlusion of the different space boundaries. The most recent developing technology of visual guidance uses only cheap sensors and attains better localization precision.

Tracking systems utilize images as input and process the camera positions to build a 3D map via multi-view geometry to allow localization and consistent map generation. Mono-tracking system [5] was the first methodology to pass the tracking system approach from robotic research into computer vision. Many tracking systems have been developed, such

as SCR-tracking system [6], DynaS [7], and DSO-Tracker [8]. These systems are designed for scenes that contain only stationary objects [9]. These systems are affected by dynamic interferences, which reduce the localization accuracy. Motion can originate incorrect obstruction of already computed features, causing path computation mistakes [8–10]. This can lead to malfunctions of wrongly ignoring dangerous objects or vehicles in the surveillance areas, causing safety issues. Increasing the precision of tracking systems in dynamic areas is a significant research area [9–11].

The objective of the proposed model is to track moving objects in dynamic environments. Therefore, the tracking system removes the dynamic environment boundaries to remove false movement information and concentrate on real moving objects in such environment boundaries. We are proposing a novel model that incorporates a combination of the deep learning method and a tracking system algorithm that rejects dynamic areas which are not within the environment boundary of interest.

The key contributions of this research are as follows.

- A tracking system that excludes dynamic environment boundaries is presented. The front end incorporates a new dynamic environment boundary detection algorithm (DRD). This detection algorithm does not affect the localization accuracy.
- The simple linear iterative clustering algorithm is enhanced to eliminate the redundant calculations and reduce the time complexity of the DRD algorithm.
- Experiments on the VID [12] and MOVSD4 [13] datasets are performed with effective results of the suggested tracking system in different dynamic settings.

This paper is structured as follows: Section 2 surveys of related work. Section 3 introduces problem definition and mathematical representation. In Section 4, experimental results are demonstrated. Section 5 depicts the conclusion and future work.

2. Related Work

Tracking systems operate on different features of objects in dynamic areas, which decreases their effect on movement estimation [14]. Tracking systems might miss dangerous moving objects in dynamic environments and might assume such objects as outliers and do not pose safety issues. Therefore, it is important to exclude the dynamic objects from the surveillance areas and concentrate on dangerously moving objects, and that is what we propose in this research. Outlier detection and rejection techniques [15–18] were employed in SCR-tracking systems. The authors in [18] utilized a robot odometer to build a dynamic feature matrix. Dynamic areas are discarded by posing restrictions on the polar and path computational geometry, the refusal outcome was influenced by the odometer accuracy. The authors in [19] proposed a method for computing static weights by merging depth information and feature static weights to approximate the camera pose. Their method eliminated the adverse impact on the pose computation that was triggered by dynamic objects. However, the previously proposed systems could not totally eradicate the effect of the motion.

Another tracking system in dynamic environments identified dynamic areas of the graph [20–22]. The authors in [20] suggested a pixel segmentation for each two consecutive video frames where the super-pixels of the current video frame are broadcast. The motion metric between the broadcast super-pixels and the previous motion is calculated, and the largest area was used to determine if the broadcast super-pixel belonged to the moving object. The authors in [21] used the variance in color hue of both the present motion area and the previous moving area to spot path variation. Pixel categorization was performed with the partition of the frame. One of the drawbacks of this algorithm is that it classifies part of the ground as a dynamic object because the ground may look like a moving object depth. Image semantics can aid the tracking system model in recognizing its surroundings [22] by optimizing features association to ensure robustness. The authors in [23] proposed a contour segmentation of the dynamic objects in the video frames using ResNet, united with a path detection algorithm to detect the dynamic status of an object. The authors in [24] employed an R-CNN for pixel-based segmentation of the dynamic background

in the video frames and prevented the tracking system from extracting dynamic features from those pixels. Moreover, dynamic points were identified with multi-view models and then detached. The occlusion was covered utilizing previous key-frame pixel information. However, deep learning models usually involve exhaustive training of the CNN with large training set requirements and high computational power. The resulting systems can be affected by the size and nature of the training data, leading to poor interpretability. A detailed comparison of recent research in tracking systems of moving objects in moving backgrounds is depicted in Table 1.

Table 1. Recent research in Tracking systems of moving objects in moving background.

Reference	Method	Description	Proposed Model	Database	Average Accuracy
[14]	Binary classification (safe/not-safe)	Tracking systems operate on different features of objects in dynamic areas, which decreases their effect on movement estimation. Tracking systems might miss dangerous moving objects in dynamic environments and might assume such objects as outliers that do not pose safety issues.	Spatial similarity map	Surveillance images database of 2500 videos	91.23%
[18]	Robot odometer to build a dynamic feature matrix	The model employed a robot odometer to build a dynamic feature matrix. Dynamic areas are discarded by posing restrictions on the polar and path computational geometry, the refusal outcome was influenced by the odometer accuracy.	Recurring CNN	7064 images of five labeled dangerous situations	90.76%
[19]	Elimination of motion areas that do not impose dangerous object movement	The method computed static weights by merging depth information and feature static weights to approximate the camera pose. The method eliminated the adverse impact on the pose computation that was triggered by dynamic objects. However, the previously proposed systems could not totally eradicate the effect of the motion.	Deep learning CNN	4970 images	92.7%
[20]	Motion estimation algorithm	The model used pixel segmentation for each two consecutive video frames where the super-pixels of the current video frame are broadcast. The motion metric, between the broadcast super-pixels and the previous motion is calculated and the largest area was used to determine if the broadcast super-pixel belonged to the moving object.	Deep CNN Architecture	6024 videos images with 30 frames each	93.4%
[20]	Depth Calculation	The model used the variance in the color hue of both the present motion area and the previous moving area to spot path variation. Pixel categorization was performed with the partition of the frame. One of the drawbacks of this algorithm is that it classifies part of the ground like a dynamic object because the ground may look as a moving object depth.	CNN and discrete cosine transform	8054 3D images	92.3%

Table 1. *Cont.*

Reference	Method	Description	Proposed Model	Database	Average Accuracy
[22]	Image semantics	The model utilizes Image semantics to aid the tracking system model in recognizing its surroundings by optimizing feature associations to ensure robustness.	Transfer learning	8192 surveillance images	91.5%
[23]	Contour segmentation	The model used a contour segmentation of the dynamic objects in the video frames using ResNet, and was united with a path detection algorithm to detect the dynamic status of an object.	Deep learning Recurring CNN model	4005 video frames	93.5% with higher CPU time
[24]	Dynamic background occlusion	The model employed pixel-based segmentation of the dynamic background in the video frames and prevented the tracking system from extracting dynamic features from those pixels.	R-CNN	4860 images	93.67%
	Our proposed model	In this study, we propose a novel model that incorporates a combination of the deep learning method and a tracking system algorithm that rejects dynamic areas which are not within the environment boundary of interest. The proposed model: Dynamic Province Reject (DPR) detects the areas of the dynamic points in the frames by epipolar feature extraction model.	epipolar feature extraction model	6608 videos	97.6%

In this study, we propose a novel model that incorporates a combination of the deep learning method and a tracking system algorithm that rejects dynamic areas which are not within the environment boundary of interest. The proposed model: Dynamic Province Reject (DPR) detects the areas of the dynamic points in the frames by epipolar feature extraction model. The proposed technique eliminates the need to process the whole video frames. The second phase of the model involves segmentation of the dynamic objects, in the involved area, utilizing a super-block low computation algorithm with disparity information extraction. The last phase is the exclusion of dynamic environment boundaries using dynamic points and object boundaries. The model uses static environment boundaries information only for pose estimation to construct accurate maps.

3. Model Description

Visual sensors, such as monocular, RGB-D, and multiple cameras, are used for the tracking system algorithm. Monocular cameras have limited sensor properties due to scale uncertainty in tracking-system-based models. RGB-D cameras produce aligned depth maps by releasing an LED light and locating its refraction. This camera has a restricted array, which is vulnerable to mistaking other light beams, such as the ones emitted by the sun. Therefore, a binocular camera was selected as a sensor for our model.

A block diagram of the proposed tracking system model is depicted in Figure 1. After obtaining the multiple camera views image, we compute the disparity parameter of all points using a local environment boundary-based block matching algorithm (LRBM). LRDM uses multiple camera views for the matching process. The detailed processing phases of our proposed model are detailed in the next subsections.

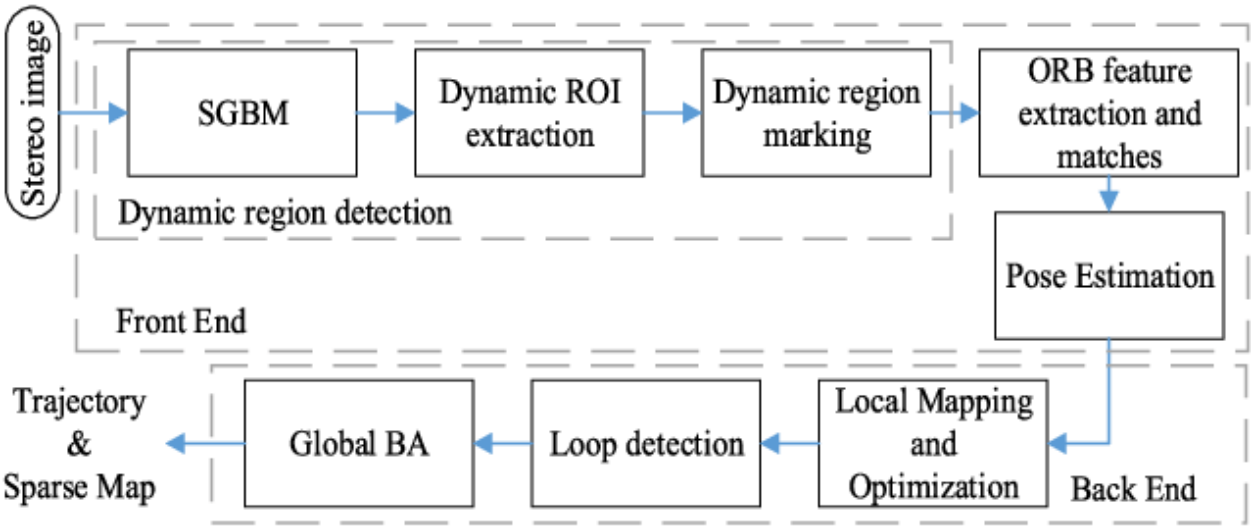


Figure 1. Proposed model framework with dynamic target detection.

3.1. Dynamic Area Extraction

The first step in the proposed DPR model is to identify moving objects in the surveillance area utilizing feature points and epipolar model constraints. Feature extraction techniques, such as scale feature transform (SIFT), accelerated feature segment test (FAST), and robust speeded-up features technique (SURF), are presented in the literature [25–27]. SIFT is described as follows:

- Sift is a rotation invariant model that displays high performance in both complex and time-consuming computation system.
- SIFT is employed to find local features in the spatial dimension.
- It detects key points and then adds descriptors to be utilized for object detection.

SURF is an enhancement over SIFT and can speed up computation. SURF is described as follows:

- SURF is a fast detection algorithm that compares the center-point's intensity level with its surrounding pixels.
- SURF is fast and robust and utilized for similarity invariant comparison of images.
- The key point of SURF algorithm is the real-time calculation of operators employing box filters for object tracking.

In Figure 2, the image corner points extraction employing the SURF algorithm is executed in OpenCV. The selected dynamic point-features are dense and stacked in a dense manner with other dynamic points. They display few to no feature points. Quadtree can be used for better detection of moving objects, dynamic feature points must be extracted from an image in a uniform way. The selection of the dynamic map in the video frame follows a uniform distribution as depicted in Figure 3. The enhanced dynamic point extraction procedure is depicted as follows.

(1) Each input frame is portioned into multiple squares. The SURF algorithm is devised to compute additional dynamic maps of each region.
(2) Each image is represented by initial nodes, and the dynamic feature points are assigned to the proper node.
(3) The algorithm tests if each node has only a single feature. If a node has more than one dynamic feature, the square including this node will be divided into four sub-squares, and a new node will be assigned to each sub-square. The dynamic feature in the old node will be assigned to the new node.
(4) Step (3) is repeated and will be stopped when the vertices that have a specific map vector achieve the required dynamic map.

Figure 2. SURF extraction algorithm (the red points are the dynamic feature point-locations).

Figure 3. SURF with Quadtree (the lines are the node borders).

Figure 3 depicts the impact of employing Quadtree in extracting dynamic feature points, resulting in a uniform distribution.

A set of homogenous dynamic feature points ($f_1(t-1), f_2(t-1), \cdots, f_n(t-1)$) of a frame $(t-1)$ is computed and extracted by the proposed method. These points are computed and tracked by the Lucas algorithm [28] and yield the corresponding dynamic

feature points ($f_1(t), f_2(t), \cdots, f_n(t)$) in the current frame t. For epipolar model, a relation between the dynamic region in $(t-1)$ and the epipolar t is calculated as follows:

$$L_i(t) = F \times f_i(t-1) \tag{1}$$

where $f_i(t-1) = [X_i(t-1), Y_i(t-1), 1]^T$ is the jth dynamic feature point of the frame $(t-1)$ and $L_i(t) = [A_i, B_i, C_i]^T$ is the epipolar line's coefficient. The jth property of $(t-1)$ is positioned in video frame t. If the scenario is not dynamic, the properties traced in frame t will be found on their corresponding node epipolar line. On the other hand, dynamic feature points that are not found on epipolar path are features of moving objects. The displacement Dis from $P_i(t)$ to its equivalent $L_i(t)$ is computed as followings:

$$Dis = \frac{|Ai \times Xi(t), Bi \times Yi(t), Ci|}{\sqrt{Ai^2 + Bi^2}} \tag{2}$$

$X_i(t)$ and $Y_i(t)$ are the coordinates of the dynamic point $f_i(t)$.

We have to differentiate between real dynamic points and noise. Therefore, a point is accepted as a property on a moving entity when the distance Dis is above a specified value V. The value V is directly proportional to the discrepancy and is calculated as follows:

$$V = k + g \times di \tag{3}$$

where, k and g are predefined constants and di is the disparity of point $P_i(t)$ in the current frame t.

To eliminate noise and other undesirable factors, an object is reflected as a dynamic object if it possesses at least four features. The disparity image obtained by LRBM is depicted in Figure 4.

(**a**) (**b**)

Figure 4. The disparity image obtained by LRBM. (**a**) Original image with feature points, (**b**) computed disparity image.

3.2. Dynamic Area Marking

The dynamic object location in the frame can be coarsely determined from the dynamic area of interest. For better division of these areas, it is essential to divide the frame and define the moving object boundary contour. Fuzzy clustering algorithms [27–29] with super-pixel division techniques can categorize the areas with the displacement similarity metric Sim. They can measure the difference between the color vector $C = [M,x,y]$ and the space vector $Z = [a,b]$. Sim is formulated as follows:

$$Sim = \sqrt{\left(\frac{\|Lj - Li\|^2}{N}\right) + \left(\frac{\|Zj - Zi\|^2}{N}\right)} \tag{4}$$

where i = [*1*,*b*] represents the class i center. j represents points inside the search area. N is the maximum spatial distance, which is determined from the image. N is defined as the preliminary spatial distance between the centers, and is calculated as follows:

$$N = \sqrt{\frac{l \times w}{b}} \tag{5}$$

where l and w are defined as the dimension of the frame, and b is size of the class locus set Z.

- The proposed phases of simple iterative clustering are defined as depicted below:
- The class locus Z is spread uniformly with N as the spatial displacement.
- The displacements between class centers and the image pixels are calculated within a 2 N × 2 N area and the pixel is allocated to the class with the least displacement.
- Each cluster's center should be updated.
- Continue looping through Steps b and c, until convergence to reach better results.

It is found that computing the displacement between the centers and its neighboring pixels is redundant. However, the algorithm has to consider the real-time requirements. Therefore, a novel cluster approach for pixel updates is proposed where unstable edge pixels are categorized and updated for each iteration.

The new update algorithm is detailed as follows:

- All the centers will be uniformly distributed and pixels near the edges will be manifested as unstable.
- The displacement between each center and the selected unstable points are calculated within a 2 N × 2 N environment boundary and the unstable points are changed.
- The center and the unstable pixels beside the unstable flag of pixels beside any unstable points are marked and updated.
- Continue looping through Steps b and c, until convergence to reach better results.

The conditions for changing the unstable point in step c can be stated as follows:

$$Flag_p = \begin{cases} 1, \; if \; Flag(p) \neq Flag(q) \\ \quad \vee Flag(q) = 1 \\ 0, \; otherwise \end{cases} \tag{6}$$

The point q represents the unstable point that is changed in step b, and p denotes the set of neighboring pixels of q. Figure 5 depicts the procedure of changing the flags for point p. Super-pixel partition is executed on the dynamic environment boundaries of interest utilizing the modified SIC technique.

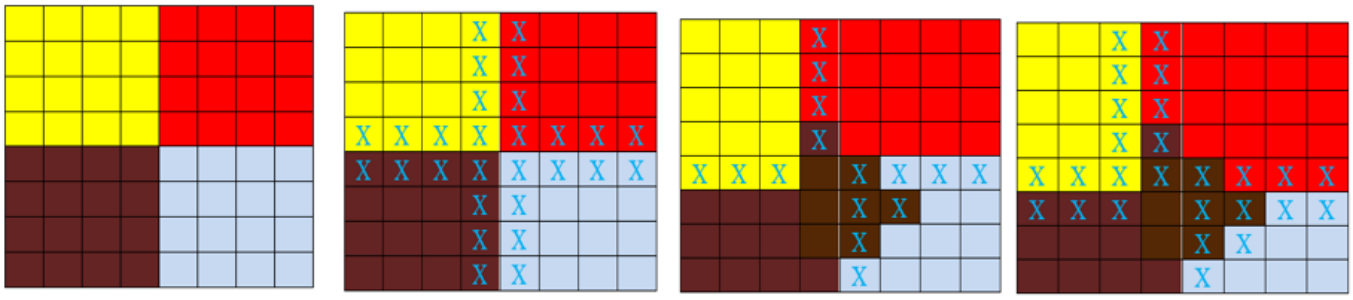


Figure 5. Procedure of changing the flags for point p.

3.3. The Tracking Model

The proposed tracking model is characterized by its robustness that utilizes a similar technique as the SCR-tracking algorithm [6]. Each frame is partitioned into non-dynamic areas. The tracking phase selects feature points in the non-dynamic area that fit the map points and computes the stance poses. Optimization of the stance is done by optimizing the projected plan errors. The mapping phase controls the handling of the map points and

executes the block motion algorithm. This phase optimizes the map points' location with respect to other involved key-frames. The loop terminating phase determines if a loop exists. In this case, it halts the mapping phase and executes a pose block motion of the map key. It can be shown in Figure 5 that the key-point matches each localization and mapping outcome. SCR-tracking uses the random sample consensus metric to reject feature points that produce a mismatch of the localization.

Dynamic Object Tracking Algorithm

The tracking of dynamic objects involves the state-space models that represent dynamic environments. To describe a dynamic environment, we need to define the state transition and the state capacity [4]. For the Bayes model of dynamic calculations, the model uses accessible information to build the posterior state density (PSD) [5]. If the system transition or capacity model is nonlinear, the PSD function will represent the non-Gaussian function [6]. The value of the state is built-in in the PSD function. An iterative method means the filter will receive data sequentially and will not store the complete data [8]. The filter comprises two phases: Prediction and alteration. The prediction phase utilizes the transition model to predict the state PSD of the next phase. The alteration phase utilizes the latest computed value to update the predicted PSD. This is attained by the Bayes distribution function of the alteration mechanism of the dynamic object state. The nonlinear filter utilizes the dynamic object state vector (OSV).

We filter and compute the OSV vector using all attainable metrics. From the viewpoint of the Bayes model, this problem is to compute a high credibility factor under a dynamic situation. Therefore, the model constructs the PSD function [13].

4. Experiments

In this section, we evaluate the effectiveness of our tracking model and describe the datasets.

4.1. Experimental Settings

The proposed model is executed using the Intel Xeon 27-d1000, CPU: i7 (3.80 GHz), GPU: Pavilion GTX 1050 8 GB, RAM: 32 GB. The simulation factors of the proposed model are depicted in Table 2.

Table 2. The simulation factors of the proposed fire detection model.

Parameter	Description
Input	256 × 256 × 3 images
Batch size	64

We utilized 70-15-15 distribution of the dataset for training, validation, and testing, respectively (Table 3). The hardware configuration used in the experiments is depicted in Table 4.

Table 3. Distribution of subsets in the experiments.

Subset	Number of Frames	%
Training subset	4340	70%
Validation subset	930	15%
Testing subset	930	15%

Table 4. The Sun station configuration.

GPU	**Eight Cores Each of 32 bits @ 1.5 GHz, 64 Gb RAM with GP I/O**
CPU	Intel Xeon processors
Operating System	UNIX System V Release 4 (SVR4)

4.2. Dynamic Environment Boundary Detection

The proposed dynamic environment boundary detection model uses the dataset described in [30]. This dataset encloses videos of dynamic targets shot of real-life scenes. The videos include dynamic environments of real-life scenes and contain moving vehicles and pedestrians. The proposed model is proven to segment the dynamic environment boundary accurately. The proposed model can differentiate between various dynamic targets even with the occlusion of the moving targets. On the other hand, other static targets are mistakenly detected as dynamic targets. The problem is the presence of many dynamic targets in the environment. This causes incorrect dynamic-feature matrix computation. This environment boundary is detected, and targets are out bounded are categorized as occlusion.

The results of this technique for the MOVSD4 and VID [13] datasets are depicted in Tables 2 and 3 as confusion matrices. The evaluation metrics of dynamic object detection are depicted in Table 5. Evaluation metrics, namely accuracy, precision, and recall, are formulated as follows [5]. The confusion matrix is depicted in Table 6. Evaluation metrics are depicted in Table 7.

$$Percision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{9}$$

where, TP is the positively detected dynamic targets, FP is the falsely detected dynamic targets, and FN is the count of the unidentified dynamic targets. The intersection ($\cap$) over the union (U) abbreviated as IU is the value of the join of the detected area of a moving target and the area of the ground truth:

$$UI = \frac{D \cap G}{D \cup G} \tag{10}$$

where, D is the determined area of the moving target and G depicts the labeled data.

Table 5. Confusion matrix for the MOVSD4 [13] dataset.

MOVSD4 [13] Dataset	**Positive**	**Negative**
Positive	3002 (TP)	18 (FN)
Negative	19 (FP)	750 (TN)

Table 6. Confusion matrix for the VID [12] dataset.

VID [12] Dataset	**Positive**	**Negative**
Positive	2120 (TP)	16 (FN)
Negative	13 (FP)	670 (TN)

Table 7. Evaluation metrics.

Dataset	Precision	Recall	Accuracy	IU
MOVSD4 [13]	97%	95.37%	97.06%	87.78%
VID [12]	98.22%	96.98%	98.3%	85.4%

As can be realized, the model proposed in this research produces worthy results for all metrics, with 97.6% precision and 96.5% recall in the dataset of MOVSD4, the metrics are a bit higher than those of VID dataset. This is because the dynamic targets in the images in MOVSD4 are closer with larger dimensions.

4.3. Evaluation of the System DPR

The DPR technique, in our research, is tested on frames (FS) with enclosed provinces of MOVSD4. FS1 is a province, and FS3, FS4, FS5, FS7, and FS8 are city provinces where FS5 and FS7 have loops. The evaluation metrics to test dynamic area rejection techniques are the path absolute difference (TAD) and pose relative difference (RPD). The TAD is computed as the change of the predicted path and the labeled paths, which depicts the DPR precision of the predicted path, and it is utilized to test the accuracy of the model. The RPD computes the change in the path measurements in the exact period, and RPD evaluates the drift. We utilized the path absolute error (TAE) as depicted in Table 8. [29], relative translation (RT), and relative rotation error (RRE) [30] for evaluations. We evaluated the proposed dynamic province rejection technique (DPR), the SCR model [6], and the DynaS model [7] for comparison, and the results are depicted in Table 5. It is observed that the SCR and DynaS models are both using multiple camera views. proposed in this research are multiple camera views. In all the experiments, we opened all the loop-closing of all the techniques. It is noted that our model has higher performance than the other models for the MOVSD4 data items that enclose dynamic objects. In these frame sequences, all objects are stationed on the corner and obstruct big regions of each sequence frame. DynaS eliminates all objects as dynamic targets of no exception, discarding some map points for locus determination. On the contrary, our model only removes moving objects. Therefore, our model outperforms the DynaS model for most sequences. We summarize the output of Table 9 into a chart for the average contrast evaluation results on MOVSD4 dataset averaging the results for all frame sequences (Figure 6).

Table 8. Contrast evaluation results in MOVSD4 dataset (Units: Meter).

Frame Sequence	SCR			DynaS			Our Proposed Method		
	TAE%	RT (Degree/100 m)	RRE (m)	TAE%	RT (Degree/100 m)	RRE (m)	TAE%	RT (Degree/100 m)	RRE (m)
FS1	1.37	0.21	5.4	1.58	0.23	3.6	0.53	0.18	3.2
FS3	0.69	0.17	0.7	0.68	0.19	4.7	0.75	0.12	0.78
FS4	0.48	0.13	0.3	0.45	0.09	1.23	0.41	0.11	0.2
FS5	0.39	0.17	0.7	0.41	0.17	0.79	0.39	0.17	0.69
FS7	0.51	0.29	0.6	0.53	0.29	0.5	0.41	0.14	0.38
FS8	1.03	0.33	3.7	1.07	0.33	3.6	0.57	0.18	2.2

Table 9. The path absolute difference (TAD) of each model for real scenes (Units: Meter).

Scene	SCR				DynaS				Our Proposed Model			
	TAE%	Mean	Median	SD	TAE%	Mean	Median	SD	TAE%	Mean	Median	SD
1	0.77	0.53	0.175	0.521	0.002	0.001	0.001	0.101	0.012	0.018	0.002	0.010
2	0.87	0.85	0.788	0.177	0.006	0.719	0.077	0.138	0.075	0.024	0.078	0.033
3	1.01	0.87	0.675	0.512	0.045	0.069	0.023	0.135	0.041	0.021	0.052	0.033
4	0.49	0.43	0.422	0.223	0.141	0.107	0.179	0.166	0.139	0.117	0.069	0.037
5	0.51	0.48	0.487	0.243	0.115	0.129	0.105	0.153	0.141	0.092	0.098	0.039
6	0.53	0.47	0.459	0.322	0.107	0.133	0.109	0.149	0.097	0.128	0.090	0.053

To validate the efficiency of the proposed model in real scenes, six scenes were considered. The first scene depicts a stationary object in a dynamic scene. The second and third scenes demonstrate a moving object in a 14-m-long line in a dynamic scene. The fourth, fifth, and sixth scenes include a moving object in a loop of 7 × 9 m in a rectangle in a dynamic scene. The dynamic settings of the first, second, and fourth scenes are of low activity, with only two moving objects. The third, fifth, and sixth scenes are characterized by high activity, with more than three moving objects at the same time. Tables 9 and 10 depict the TAD and RPD (translation) of each model for the six scenes. The path absolute error (TAE), mean and median errors, and standard deviation are depicted in those tables. The path absolute error (TAE) is susceptible to large errors and can test the model robustness [31–33], whereas standard deviation usually tests the system stability [34,35]. These results are averaged in Figures 7 and 8.

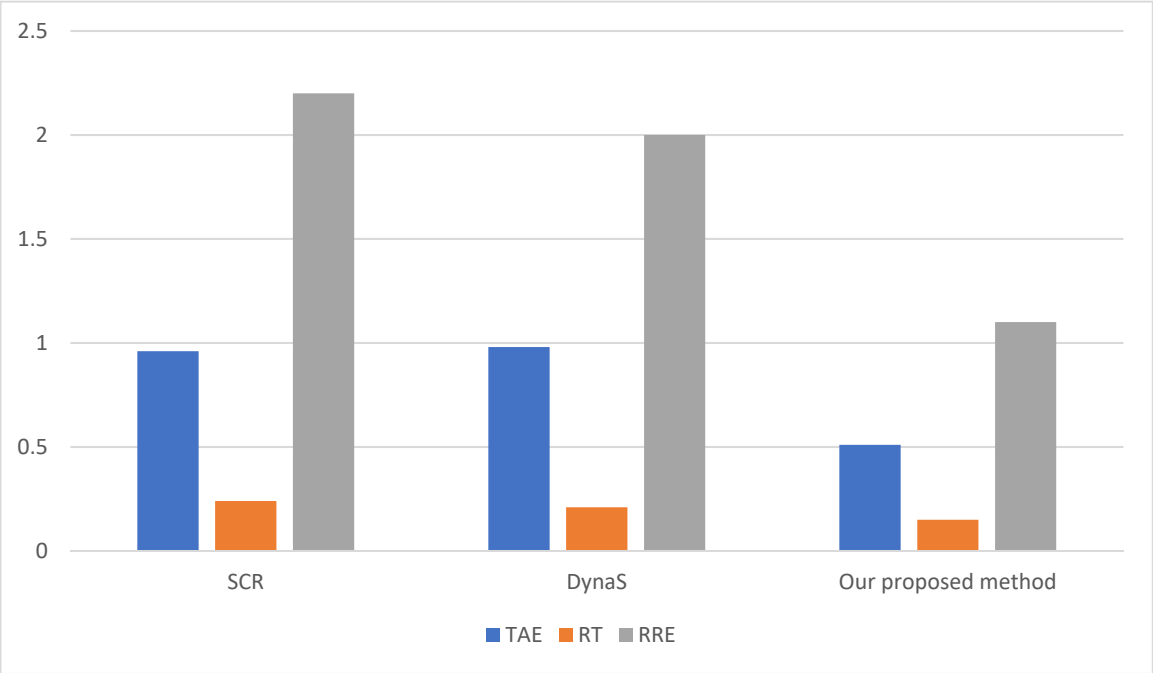


Figure 6. Average contrast evaluation results in MOVSD4 dataset averaging the results for all frame sequences.

Table 10. The RPD (Translation) of each model for real scenes (Units: Meter).

Scene	SCR				DynaS				Our Proposed Model			
	TAE%	Mean	Median	SD	TAE%	Mean	Median	SD	TAE%	Mean	Median	SD
1	0.013	0.008	0.008	0.012	0.003	0.003	0.001	0.001	0.010	0.008	0.002	0.011
2	0.072	0.014	0.014	0.013	0.006	0.019	0.017	0.018	0.013	0.014	0.078	0.013
3	0.041	0.021	0.011	0.013	0.045	0.039	0.013	0.015	0.023	0.011	0.052	0.013
4	0.039	0.017	0.014	0.010	0.041	0.017	0.079	0.006	0.017	0.017	0.019	0.007
5	0.041	0.014	0.014	0.013	0.015	0.029	0.015	0.005	0.023	0.022	0.018	0.019
6	0.067	0.027	0.021	0.043	0.017	0.033	0.019	0.009	0.030	0.028	0.010	0.013

As can be perceived, the proposed model outperforms DynaS in real scenes. Our model also performs better than SCR. In the first scene, where the camera is static, the proposed model is unaffected by the dynamic object. In the second to sixth scenes, the localization precision of the model is better than SCR, where the mean enhancement path absolute error (TAE) and S.D. are 83% and 81% for TAD and 33% and 42% for RPD, respectively. The third and fifth scenes depict that the proposed model has higher performance in extreme scenes, which include movements. The enhancement of the proposed model, over the SCR model in the MOVSD4, is less than in the real-life scenes. This is due to the broad view that is utilized in MOVSD4 and the low rate of moving targets.

Figure 9 depicts the results of the path absolute inaccuracy between our model and the SCR model in the first scenario. The moving object moves from the center of frames 122, 213, and 225 and then disappears at frame 399. The path of the SCR model drifts to the right if the moving target moves and then goes to the far right when the moving target disappears. The path absolute error is evened out at 1.17 m after the moving target disappears. On the contrary, our proposed model is unaffected by the entire movement of the target. Figure 10 depicts the count of dynamic points that are excluded in the first scene frames. In this experiment, we compute 1300 points for each frame. These excluded points are replaced by points in the non-dynamic areas to include enough feature points for mapping.

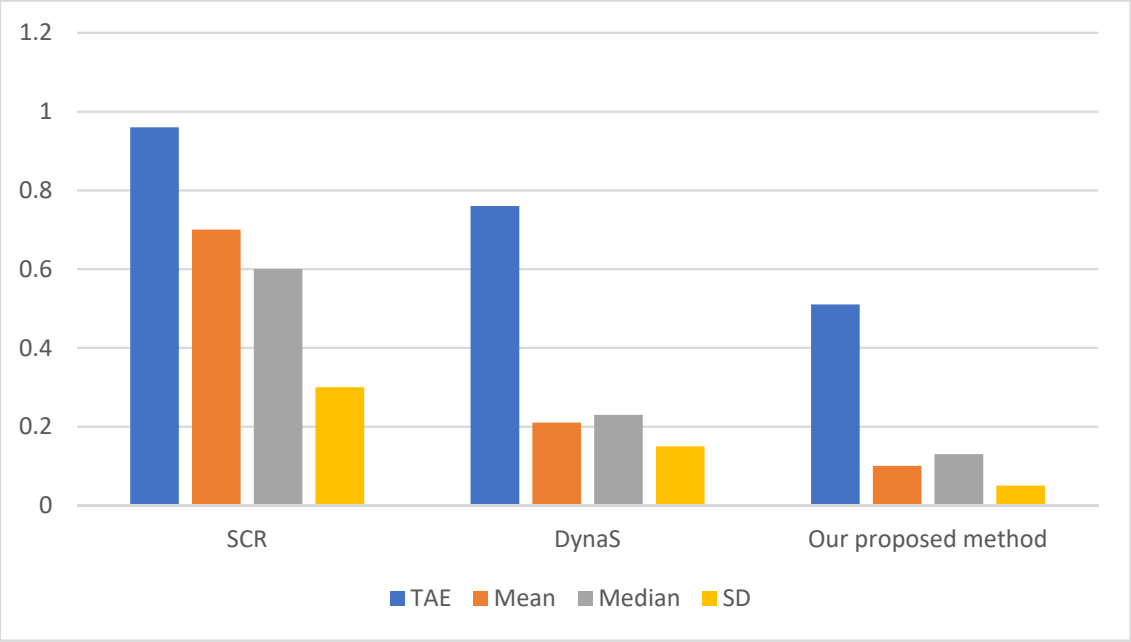


Figure 7. The average path absolute difference (TAD) of each model for real scenes.

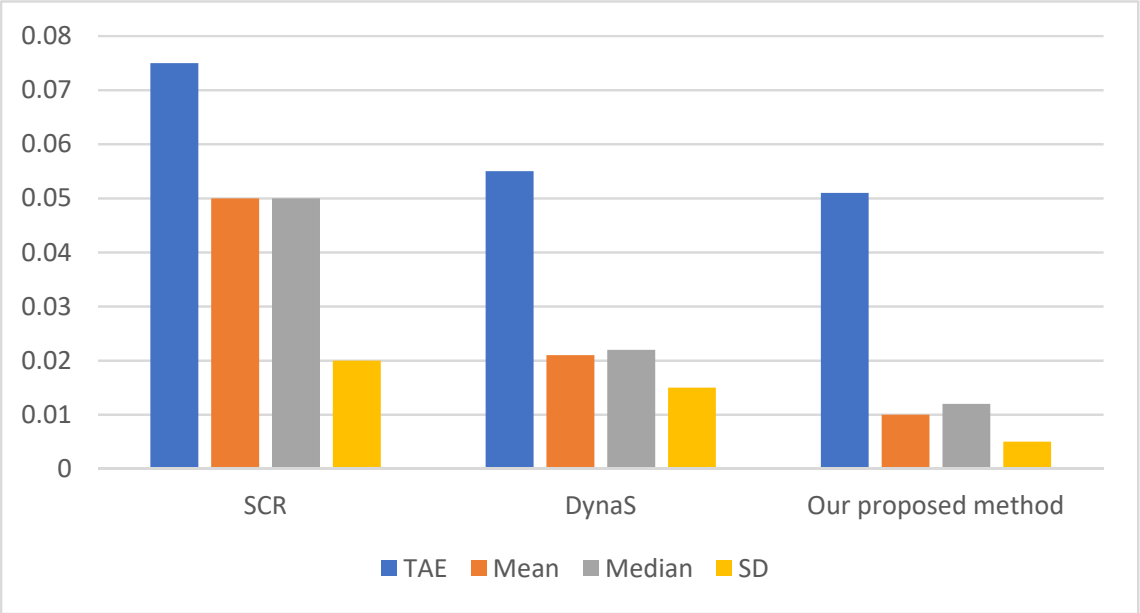


Figure 8. The average RPD (Translation) of each model for real scenes.

Figure 11 depicts the expected trajectories and the ground truths of the SCR, DynaS, and the proposed model for the second to sixth scenes.

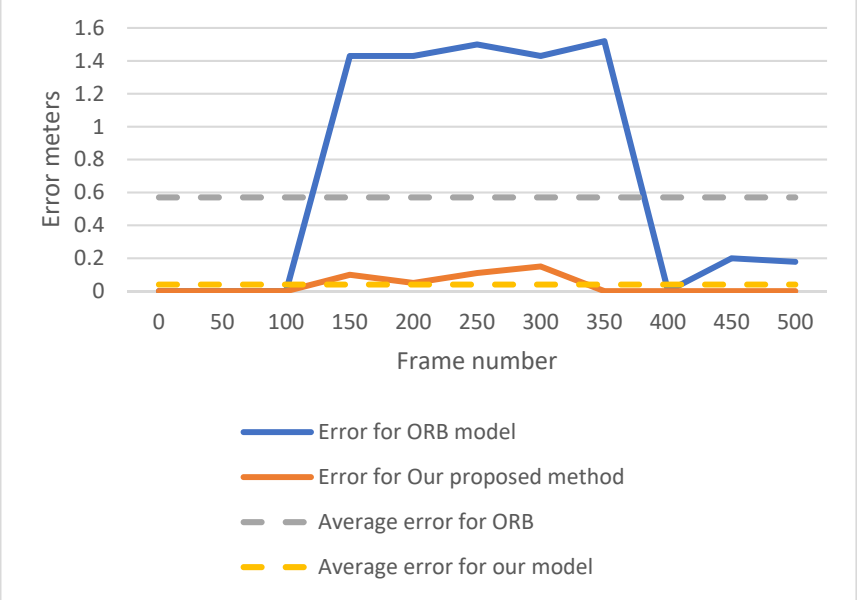


Figure 9. Path absolute error of each model for the first scene. (**a**) Frame 145, the dynamic object was moving to the right towards the camera. (**b**) Frame 254, the dynamic object was turning. (**c**) Frame 352, the dynamic object was moving to the other direction.

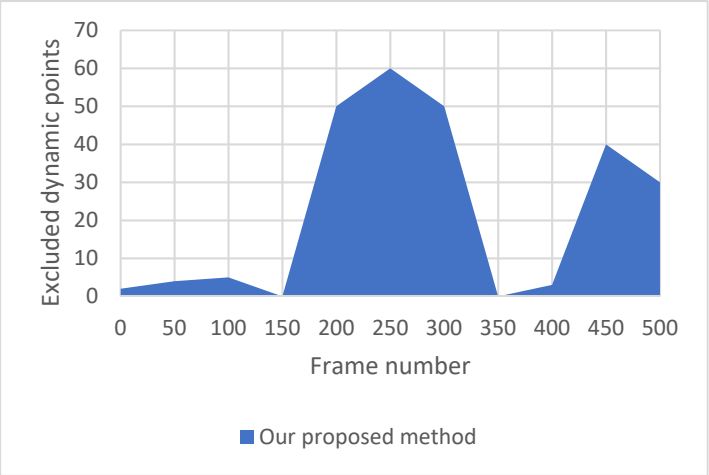
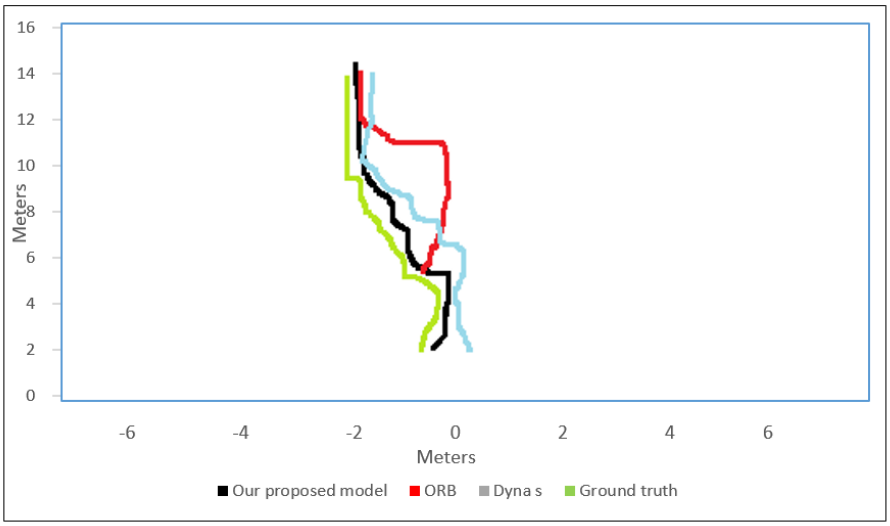


Figure 10. The count of excluded dynamic points by our model in the first scene. Our model performs well and fails only when the dynamic object turns (frame 243–270) or when the dynamic object is slow (frame 452–500).

Table 11 depicts the mean time consumed in real-life scenes for the dynamic object detection algorithm of each model. The runtime of DynaS model, which utilizes a Region-CNN in the segmentation algorithm, is computed by executing the algorithm on an Nvidia GTX 560 GPU, while our model can attain 8.4 frames per second in real-life settings and attains the real-time requirements.

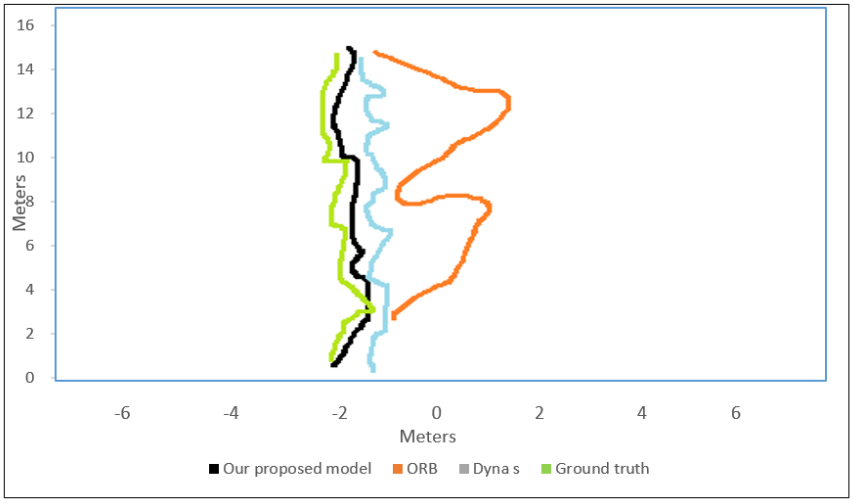
Table 11. Runtime for the dynamic object detection algorithm of Each model (Unit: ms).

	Dynamic Segmentation Time	Tracking Time	The Dynamic Object Detection
SCR model	Does not have segmentation	48.3	48.3
DynaS model	501.4	77.3	578.7
Our proposed model	83.4	51.2	134.7

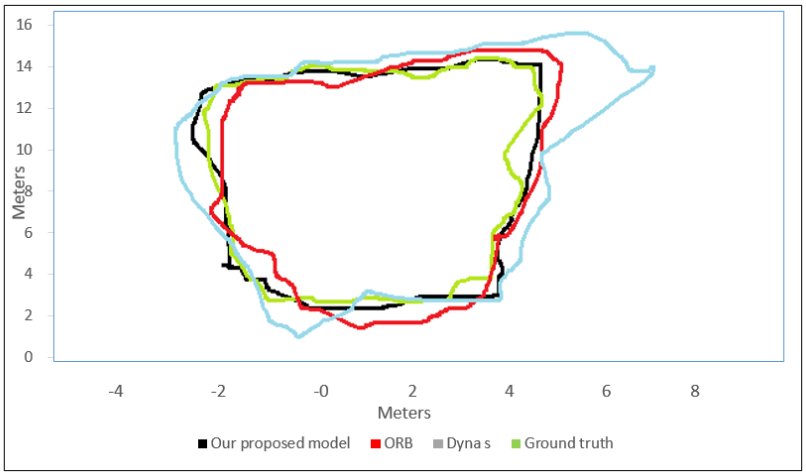


(**a**) Scene 2

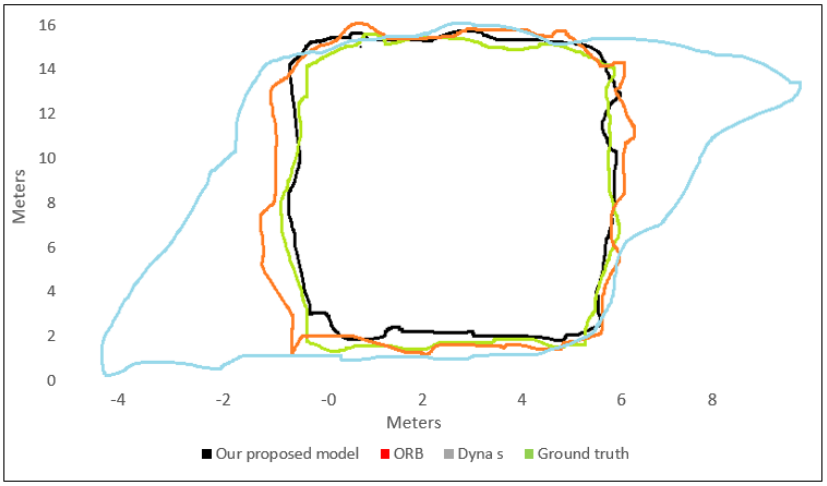
Figure 11. *Cont.*



(**b**) Scene 3



(**c**) Scene 4



(**d**) Scene 5

Figure 11. Predicted trajectories versus ground truth in different scenes for the compared models.

5. Conclusions

In this research, we proposed a wrongly moving target detection model. The moving regions are excluded by employing dynamic-feature map computation. The map was calculated from the path curves by following the motion from the local environment boundary-based block matching algorithm (LRBM) in consecutive frames. The borders were categorized utilizing the discrepancy features of the super-blocks, incorporating a feature map to determine complex areas. The dynamic province rejection technique (DPR) utilized the data in the non-moving region to predict the path to eradicate the impact of the dynamic objects on the model and enhance the mapping accuracy. Experiments were performed on two datasets in both the VID and the MOVSD4 datasets. The results clarified the accuracy of our dynamic region detection technique. The proposed wrongly moving target detection model outperformed the SCR in both datasets. Our proposed model's performance is also higher than DynaS model with better speed (7.4 frame per second). Finally, the proposed model can enhance localization accuracy in dynamic settings and attain real-time requirements.

However, our model has some limitations. First, our model attained less performance when dynamic targets were moving in opposite directions. Second, constraints in the model depend on experimentations to attain higher performance. Therefore, future work can propose trajectory prediction for objects moving in opposite directions and for objects that appear and disappear quickly.

Author Contributions: Conceptualization, H.A.H.M. and N.A.H.; methodology, H.A.H.M.; software, H.A.H.M.; validation, H.A.H.M., N.A.H. and A.A.A.; formal analysis, H.A.H.M.; investigation, H.A.H.M.; resources, H.A.H.M.; data curation, H.A.H.M.; writing—original draft preparation, H.A.H.M.; writing—review and editing, H.A.H.M.; visualization, H.A.H.M.; supervision, H.A.H.M.; project administration, H.A.H.M.; funding acquisition, H.A.H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R113), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
2. Collins, R.; Zhou, X.; Teh, S.K. An open source tracking testbed and evaluation web site. *IEEE Workshop Perform. Eval. Track. Surveill.* **2020**, *2*, 35.
3. Xu, H.; Yang, M.; Wang, X.; Yang, Q. Magnetic sensing system design for intelligent vehicle guidance. *IEEE/ASME Trans. Mechatron.* **2020**, *15*, 652–656.
4. Loevsky, I.; Shimshoni, I. Reliable and efficient landmark-based localization for mobile robots. *Robot. Auton. Syst.* **2021**, *58*, 520–528. [CrossRef]
5. Akter, S.; Habib, A.; Islam, M.A.; Hossen, M.S.; Fahim, W.A.; Sarkar, P.R.; Ahmed, M. Comprehensive Performance Assessment of Deep Learning Models in Early Prediction and Risk Identification of Chronic Kidney Disease. *IEEE Access* **2021**, *9*, 165184–165206. [CrossRef]
6. Murrtal, R.; Tardos, J. SCR: An open-source SLAM system for monocular stereo and RGB-D cameras. *IEEE Trans. Robot.* **2021**, *33*, 1255–1262. [CrossRef]
7. Bescos, A.; Facil, J.; Neira, J. DynaS: Tracking mapping and in painting in dynamic areas. *IEEE Robot. Auton. Lett.* **2019**, *3*, 4076–4083. [CrossRef]
8. Ronzoni, D.; Olmi, R.; Fantuzzi, C. AGV global localization using indistinguishable artificial landmarks. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 287–292.
9. Hafez, R.; David, J. SLAM2: A SLAM system for monocular stereo and RGB-D cameras. *IEEE Trans. Robot.* **2020**, *33*, 1255–1262.
10. Mime, J.; Bayoun, D. LSD: Large static direct monocular model. *Comput. Vis.* **2020**, *7*, 83–89.
11. Ahmed, M.; Cremers, D. Indirect deep learning odometer model. *IEEE Trans. Trans. Pattern Anal. Mach. Intell.* **2019**, *4*, 61–65.

12. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The VID dataset. *J. Robot. Reason.* **2021**, *32*, 123–127.
13. Fisher, R. The MOVSD4 surveillance ground-truth data sets. In Proceedings of the IEEE Workshop Performing Evaluation Tracking Surveillance, Cairo, Egypt, 17–19 December 2019; pp. 12–17.
14. Fuentes, J.; Ascencio, J.; Mancha, J. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev.* **2019**, *43*, 55–81. [CrossRef]
15. Saputra, M.; Markham, A.; Trigoni, N. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Comput. Surv.* **2020**, *51*, 37. [CrossRef]
16. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y. Past present and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
17. Wang, Y.; Lin, M.; Ju, R. Visual SLAM and moving-object detection for a small-size humanoid robot. *Adv. Robot. Syst.* **2021**, *7*, 133–143. [CrossRef]
18. Kundu, A.; Krishna, K.; Sivaswamy, J. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 436–441.
19. Li, S.; Lee, D. RGB-D SLAM in dynamic environments using static point weighting. *IEEE Robot. Autosomes* **2020**, *2*, 223–230. [CrossRef]
20. Tan, W.; Liu, H.; Bao, H. Robust monocular SLAM in dynamic environments. In Proceedings of the 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, SA, Australia, 1–4 October 2013; pp. 209–218.
21. Fischler, M.; Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **2021**, *24*, 381–395. [CrossRef]
22. Supreeth, H.S.G.; Patil, C.M. Moving object detection and tracking using deep learning neural network and correlation filter. In Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 20–21 April 2018; pp. 1775–1780. [CrossRef]
23. Alcantarilla, P.; Yebes, J.; Almazan, J.; Bergasa, L. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 190–197.
24. Giordano, D.; Murabito, F.; Spampinato, C. Superpixel-based video object segmentation using perceptual organization and location prior. *Comput. Pattern Recognit.* **2020**, *6*, 484–489.
25. Hu, M.; Liu, Z.; Zhang, J.; Zhang, G. Robust object tracking via multi-cue fusion. *Signal Process.* **2017**, *139*, 86–95. [CrossRef]
26. Yu, C.; Liu, Z.; Wei, Q. DS-SLAM: A semantic visual SLAM towards dynamic environments. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1168–1174.
27. Machiraju, G.S.R.; Kumari, K.A.; Sharif, S.K. Object Detection and Tracking for Community Surveillance using Transfer Learning. In Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 20–22 January 2021; pp. 1035–1042. [CrossRef]
28. Hirschmuller, H. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *30*, 328–341. [CrossRef]
29. Lowe, D. Distinctive image features from scale-invariant key points. *J. Comput.* **2020**, *6*, 91–110.
30. Bay, H.; Ess, A.; Gool, L.V. SURF: Speeded up robust features. *Proc. Conf. Comput. Vis.* **2021**, *3*, 346–359.
31. Rosten, E.; Porter, R.; Drummond, T. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach.* **2021**, *32*, 105–119. [CrossRef]
32. Achanta, R.; Süsstrunk, S. SLIC super pixels compared to state-of-the-art super pixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *4*, 227–234.
33. Sturm, J.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 573–580.
34. Kerl, C.; Sturm, J.; Cremers, D. Robust odometry estimation for RGB-D cameras. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3748–3754.
35. Sun, Y.; Meng, M. Motion removal for reliable RGB-D SLAM in dynamic environments. *Robot. Auton. Syst.* **2018**, *10*, 115–128. [CrossRef]