applied sciences

MDPI

Article

Enhancement of Multi-Target Tracking Performance via Image Restoration and Face Embedding in Dynamic Environments

Ji Seong Kim, Doo Soo Chang and Yong Suk Choi *

Artificial Intelligence Laboratory, Hanyang University, Seoul 04763, Korea; intelli8786@hanyang.ac.kr (J.S.K.); tim0225@hanyang.ac.kr (D.S.C.)

* Correspondence: cys@hanyang.ac.kr

Abstract: In this paper, we propose several methods to improve the performance of multiple object tracking (MOT), especially for humans, in dynamic environments such as robots and autonomous vehicles. The first method is to restore and re-detect unreliable results to improve the detection. The second is to restore noisy regions in the image before the tracking association to improve the identification. To implement the image restoration function used in these two methods, an image inference model based on SRGAN (super-resolution generative adversarial networks) is used. Finally, the third method includes an association method using face features to reduce failures in the tracking association. Three distance measurements are designed so that this method can be applied to various environments. In order to validate the effectiveness of our proposed methods, we select two baseline trackers for comparative experiments and construct a robotic environment that interacts with real people and provides services. Experimental results demonstrate that the proposed methods efficiently overcome dynamic situations and show favorable performance in general situations.

Keywords: computer vision; multiple object tracking; online object tracking; image restoration; data association; visual embedding

check for updates

Citation: Kim, J.S.; Chang, D.S.; Choi, Y.S. Enhancement of Multi-Target Tracking Performance via Image Restoration and Face Embedding in Dynamic Environments. *Appl. Sci.* **2021**, *11*, 649. https://doi.org/10.3390/app11020649

Received: 30 July 2020
Accepted: 16 September 2020
Published: 11 January 2021

1. Introduction

The multiple object tracking (MOT) problem aims to assign IDs to multiple detected targets and to estimate the trajectory of the object until each tracking target disappears. Recently, high-performance real-time MOT research studies are required for scenarios such as human-computer interaction, autonomous vehicles, and humanoid robots. For this reason, researches for improving real-time MOT performance such as [1–5] have been actively conducted. Existing MOT frameworks can be classified into two types, offline and online, depending on the temporal range of data to be considered [6]. The offline methods [7–9] consider the range from the past to the future, while the online methods [10–14] consider the range from the past to the present. In general, the offline methods perform better than the online methods by global optimization considering the future state, but they are not suitable for real-time tracking applications such as the previous scenario examples. The online MOT frameworks for real-time tracking are often applied in complex, dynamic, or unexpected situations, but overcoming tracking failures in these environments remains a challenge.

Online MOT frameworks need the best data association in every frame because only current and past frames are considered. In the MOT problem, the data association generally means updating the state of an object being tracked by the collaboration of a motion model and an appearance model. The motion model compares the positional similarity between the current state of the tracking object and the current detection result. In this case, prediction methods such as Kalman filters [15,16] or particle filters [17,18] are used to predict the current position of the tracking object, and the appearance model compares the similarity of the appearance between the past state of the tracking object and the present detection result. In this case, a method such as visual embedding is used to effectively extract features from a noisy image.

The dynamic movements of a tracking object and a camera in a real-time environment cause the reliabilities of the motion model and the appearance model to decrease. Motion models are negatively impacted by the complexity of moving the camera and objects separately. This is because the motion models of existing frameworks generally adopt linear functions, making it difficult for them to infer objects with nonlinear movements. Appearance models are negatively affected by motion blurs caused by dynamic movements. Because the surrounding pixel information is mixed, the detailed pixel representation, especially the texture information of the picture, is lost, and the outline of the object is blurred. This causes difficulties in distinguishing the boundary between the background and the object. The motion blur refers to a phenomenon in which pixels bleed due to the movement of the photographing object while the photosensor of the camera records an image [19–21]. The phenomenon occurs often when there is vibration caused by an uneven road surface, when the tracking target moves quickly, and when the camera mounted on the moving platform moves. As a result, the dynamic situation during multiple object tracking negatively affects the motion model and the appearance model.

In this paper, we propose three methods to overcome the limitations of existing multiple object tracking in the event of dynamic movement. The first is to perform re-detection after the image restoration on the detection result whose reliability is lowered by noise. This makes it possible to calibrate ambiguous detection results that the detector could not screen. The second is to classify and restore the damaged area before the image is entered into the appearance model. This makes it easy to guess the intact state of a damaged image and match the state of the existing object that the appearance model remembers with the current detection of the state change. We adopt and train a GAN (generative adversarial networks)-based image inference model to recover damaged images due to dynamic situations in the previous two approaches. The third method introduces a face appearance model, which is an association method that uses face features. This improves the performance of the discrimination using a large amount of information on the face and a relatively low number of occlusions.

Many MOT studies use the MOT Challenge [22] Benchmark dataset to evaluate the performance of the framework. However, since it targets a stationary or smoothly moving environment, it is difficult to simulate an environment (e.g., robot, car) in which real-time MOT is applied. We thus constructed a robot environment that can provide services in a real-time environment and produced the images observed from the robot viewpoint as a benchmark set following the MOT16 benchmark rule.

Our main contributions in this paper are as follows.

1. We present three methods to enhance the performance for multiple object tracking in a dynamic environment. Those three methods overcoming the dynamic situation contributes to improved detection, improved identification, and a lower chance of association failures, respectively.
2. Since each of the proposed methods has modularity, there is no cost for re-learning the entire framework, so it can be easily attached to various trackers.
3. To demonstrate the effectiveness of the proposed methodology, we constructed a benchmark set on a real robot environment and verified our approaches through experimental ablation studies.

2. Related Work

Online multiple object tracker. Strong motion models and appearance models are essential for online MOT methods due to the consideration of the optimal selection in the current frame without future frames. With the advent of the latest advanced object detectors [23–25], various MOT methods that link their tracking based on detection results have become popular. The work in [26] proposed a simple motion model based on a Kalman filter affected by the performance of the latest CNN (convolutional neural network)-based object detector. Furthermore, [27] proposed a model using a CNN-based appearance model for [26]. The authors in [28] proposed a method of classifying the correct detection candidates among the crowd and selecting the optimal detection candidate using the heatmap generation model. [29] adopted an RNN (recurrent neural network) to cope with the problem of the occlusion between objects by integrating the spatial and temporal information. The work in [30] improved model performance by integrating the information from the CNN intermediate layer to compensate for the information loss that existing tracking frameworks result from using only the last CNN information in a detector. In the situations of providing real-time services, dynamic or unexpected movements frequently occur, but the existing MOT methods do not often cope with such situations resulting in the poor associations. On the contrary, we apply a method to overcome the dynamic situations to the traditional online MOT frameworks and evaluate its effectiveness in the experiments.

GAN-based image inference model. Generative adversarial networks (GAN) [31] propose an adversarial loss in which two competitors, discriminator and generator, compete and learn from each other. Deep Convolutional GAN (DCGAN) [32] designed a CNN-based generator for image inference using a GAN. It indicates that when using the adversarial loss for the image inference problem, the pixel distribution close to the actual data can be obtained, resulting in more realistic images compared to the autoencoder-based model. As a result, great progress in the image inference problem such as the style transfer [33–35], the super resolution [36], and deblurring [37]. In this paper, we implement an image restoration module for MOT by adopting and learning a GAN-based image inference model for the damaged image restoration.

Appearance embedding model. Identifying whether two images represent the same person or not is accompanied by considerable difficulties due to the curse of dimensionality. In particular, identifying an unaware person adds to the difficulty. To overcome these problems, image embedding methods [38,39] using CNN were proposed recently. These models can learn to represent the whole body or a part of the body as feature vectors with small dimensions, and after learning, they are able to extract features of people that are not involved in the learning. In particular, [39] proposed a learning method using the triplet loss which achieves great results in facial recognition. We try to use a face feature to relax the problem that occurs when only a body feature is used to implement the identification function. There are two problems with using body features only. Firstly, because the boundaries become blurred when the occlusion between multiple people occurs, the embedding results mixed with the features of several people can be extracted. Secondly, the objects may be wearing similar clothes, which leads to less differentiation. To alleviate these limitations, we propose an appearance model using face features with a low incidence of occlusion and high discrimination.

3. Method

Our main goal is to overcome association failures caused by dynamic situations when executing MOT in real-time. Figure 1 shows the overall structure of our framework as data flows. In this section, we propose three strategies to achieve our goals in the order of data flow. The following summarizes each of the methods we propose.

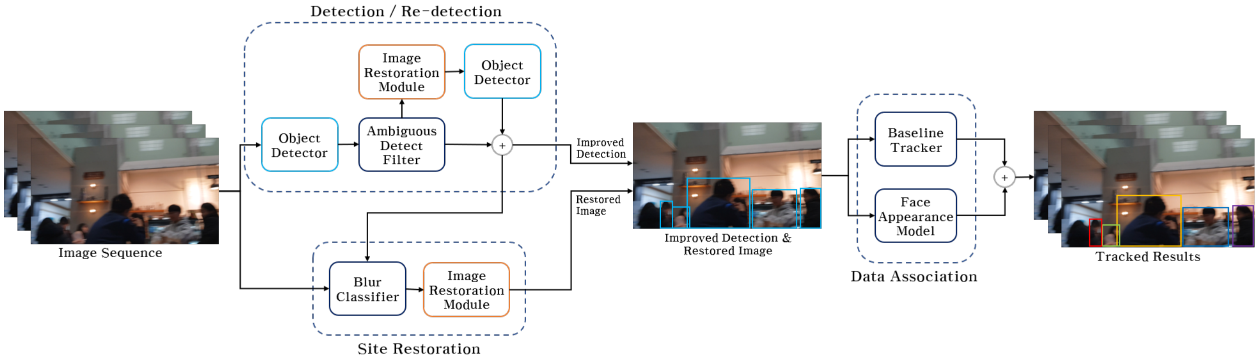


Figure 1. A schema of the proposed model. It illustrates the integration and work flow of our three methods.

Section 3.1 Re-detection: Re-detection is performed in the detection/re-detection process. This section describes the process of defining ambiguous detection results and re-detecting them after image restoration to increase the reliability of the detection results.

Section 3.2 Site restoration: This section presents the site restoration process of defining the noisy area and restoring the image of such an area to improve the reliability of the image to be used in the appearance model.

Section 3.3 Face appearance model: the face appearance model is performed in the data association process, and defines how to associate the appearance model that uses face features to overcome connection failures of baseline trackers.

3.1. Re-Detection

Since the online MOT framework does not consider the future state, the best choice is needed at every frame. Since the candidate for the association is generally proposed from the detection result, it has a high dependency on the detection result. Therefore, if the false detection on an object from an image with noise can be reduced, the tracking on the wrong object and the loss of the tracked object due to the detection failure can be prevented. Figure 2 illustrates our re-detection method. It aims to increase the detection reliability by re-detection after reconstructing the image for the ambiguous detection results that are not too low or not high enough.

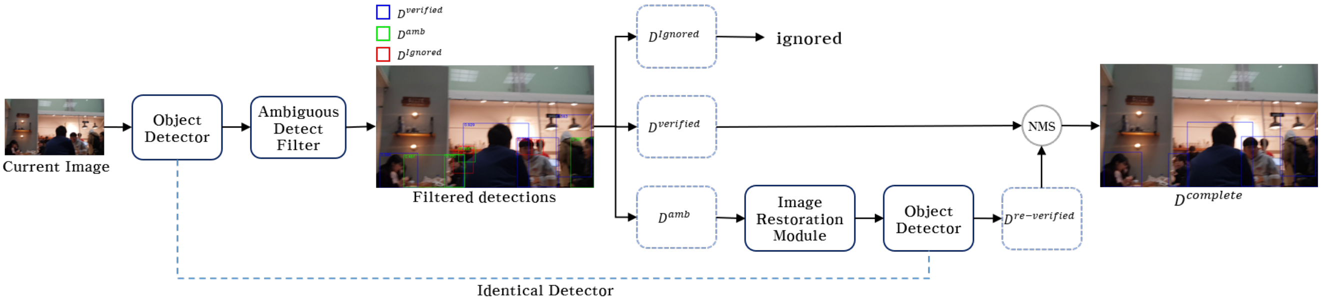


Figure 2. Schematic representation of the proposed re-detection method. It illustrates how the re-detection method works according to the data flow.

Firstly, the raw detection result is required to classify the ambiguous detection result. The raw detection result D^{origin} of the current input image x can be obtained using the pre-trained HumanDetector as follows.

$$D^{origin} = \{d_1, d_2, \ldots, d_{detectsNum}\} \leftarrow HumanDetector(x) \tag{1}$$

The variable d means a detected instance which includes the location information (x, y, w, h) and the confidence c^{body}, thus, $d = (b^{detect}, c^{body})$. Here, b^{detect} means the x,y coordinates, area, and height values that make up the bounding box, and c^{body} means the reliability of the detected object.

To classify an ambiguous detection set, D^{origin} should be separated into an ambiguous detection set D^{amb} and a verified detection set $D^{verified}$ depending on whether re-detection is needed or not, respectively. We define the confidence threshold, τ^{detect} and τ^{amb} to classify detection results with the confidence that is not too low or not too high enough. τ^{detect} stands for the most basic threshold for the detection, and τ^{amb} is a threshold to find ambiguous detections by ignoring low confidences. This process is defined as ambiguous detect filtering and is formulated as:

$$D^{verified} = \left\{ d \mid c^{body} \geq \tau^{detect}, d \in D^{origin} \right\} \tag{2}$$

$$D^{amb} = \left\{ d \mid \tau^{detect} > c^{body} \geq \tau^{amb}, d \in D^{origin} \right\} \tag{3}$$

If the value of the confidence c^{body} is lower than τ^{amb}, the target is determined to be a non-object, thus it is excluded from the detection set and not used in the tracker.

For the classified ambiguous detection set D^{amb}, the re-detection set $D^{redetect}$ can be obtained using the following definition. It functions to restore the ambiguous detection regions and then re-detect them by reusing existing detectors. The RestorationModule used to restore damaged images uses a GAN-based image inference model. The detailed procedure of training the module to optimize our model is described in Section 4.1.

$$D^{redetect} \leftarrow HumanDetector(RestorationModule(Crop(x_{current}, D^{amb}))) \tag{4}$$

To classify reliable detections from the re-detection result set $D^{redetect}$, we use the following definition. It classifies the re-verified detection using the detection threshold $\tau^{redetect}$ for the confidence $c^{redetect}$.

$$D^{re-verified} = \left\{ d \mid c^{redetect} \geq \tau^{redetect}, d \in D^{redetect} \right\} \tag{5}$$

As a result of these processes, a combination of $D^{verified}$ and $D^{re-verified}$ can be used to construct a detection set $D^{complete}$ to be used for the tracking association. However, since the re-detected object may indicate the same object as the existing detection result, there is a possibility that the duplicate objects exist in $D^{complete}$. Therefore, NMS (non-maximum suppression) is performed to remove the redundancy after constructing the union. The following definition refers to the NMS process, wher τ^{nms} is the IOU (intersection over union) threshold for the NMS.

$$D^{complete} = NMS(D^{verified} \cup D^{re-verified}, \tau^{nms}) \tag{6}$$

The final detection set $D^{complete}$ is used as a set of candidates for restoring the blur sites in the Section 3.2.

3.2. Site Restoration

Frameworks for the online MOT problem generally require one or more powerful appearance models. The appearance model aims to determine that a tracklet (an object being tracked) and a detected object are the same object based on their visual information. The appearance model is mainly used when the motion model cannot predict due to the complicated movement of the object, or when re-identification is needed based on the visual data of the object due to the failure of the detection. However, unpredictable noise such as motion blur caused by the movement of an object has a negative effect on the data association because of difficulties in the identification of the appearance model. We thus discriminate whether the detected objects are blurry images or sharp images before the image is used in the appearance model, and then restore the detection regions of those discriminated as blurry images. Figure 3 represents our site restoration method.

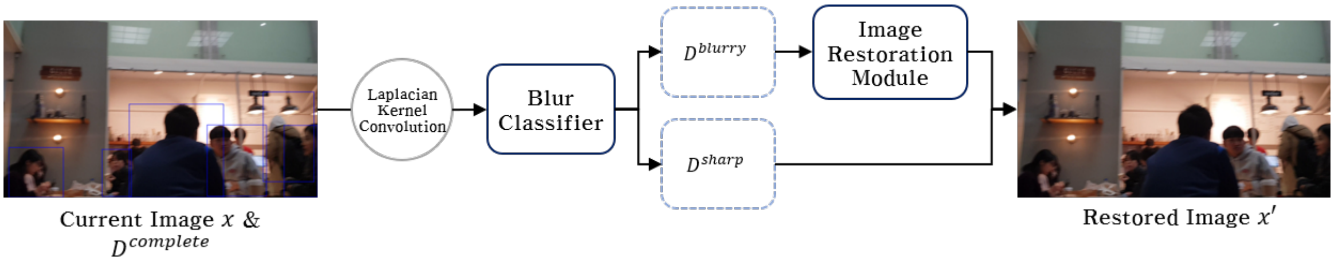


Figure 3. Schematic representation of the proposed site restoration method.

To classify whether each element of $D^{complete}$ is blurry or sharp, we use the Laplacian kernel, inspired by [40]. Laplacian kernels are generally used to detect edges of an image, but can be used to quantify the blur of an image as well. The usability is based on the fact that sharp images show a large number of edge detections. On the contrary, blurry images show a small number of edge detections. The variance value of the convolution operation using a Laplacian kernel is called Laplacian variance, which means the quantified blur. Therefore, we can discriminate that the Laplacian variance value is blurry at when low or and sharp at when high.

We define the function to find Laplacian variance of the detection set D as:

$$L = \{l_i \mid i = 1, 2, \ldots, detectsNum\} \leftarrow LaplacianVariance(GrayScale(x), D) \tag{7}$$

If the Laplacian variance is lower than the blur threshold τ^{blur}, d is determined to be thea blurry image.

$$D^{blurry} = \left\{d \mid l \leq \tau^{blur}, d \in D^{complete}\right\} \tag{8}$$

Finally, the image restoration is conducted on the blurry image using the *Restoration Module*. Consequently, in order for the reconstructed image to be used in the appearance model, the damaged areas of the original image are replaced with the reconstructed image.

3.3. Face Appearance Model

When tracking a person, the face of the target can be observed in many situations, allowing face data to be used for identification. The face data is advantageous for identification compared to the other recognizable information of the body. For example, it is less likely to meet people with similar faces than to meet people with similar fashions. In addition, features are less likely to be mixed because the possibility of occlusion is relatively less than when using the whole body. We propose an appearance model that uses face features to compensate for the problems that arise when using only body features. Figure 4 depicts our proposed face appearance model.

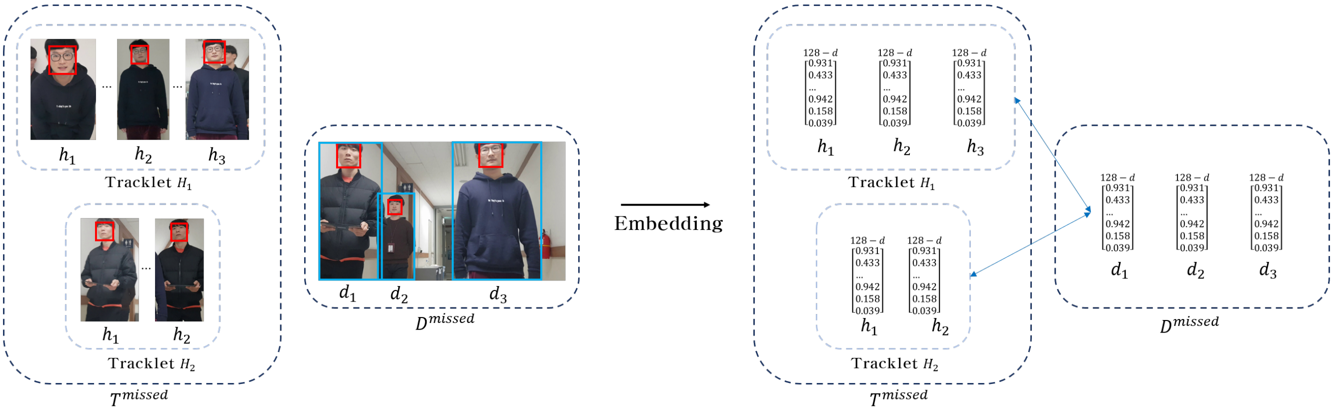


Figure 4. Schematic representation of the proposed face appearance model.

Our association method aims to associate candidates that are not yet associated after executing that of the baseline tracker. The unassociated detection set is defined as $D^{missed} = \{d_1, d_2, \ldots, d_{missedDets}\}$ and has the detection position d as an element. The unassociated tracklet set is defined as $T^{missed} = \{H_1, H_2, \ldots, H_{missedTrks}\}$ and has a tracklet H as an element. Since the tracklet has its past trajectories, it is defined as $H = \left\{h_1, h_2, \ldots, h_{historyNum}\right\}$ where h is the past position as an element.

For the extraction of face features, the following definitions are used to detect and embed the candidates' faces. At this time, the embedding is performed only for those whose detected face confidence are greater than or equal to $\tau^{faceDetect}$ as follows:

$$b^{face}, c^{face} \leftarrow FaceDetector(Crop(x, d)) \tag{9}$$

$$v \leftarrow FaceEmbeddingModule(Crop(x, b^{face})) \quad if \quad c^{face} \geq \tau^{faceDetect} \tag{10}$$

Consequently, the feature vector v_d is obtained from the detection position d, and the feature vector set V_H is obtained from the tracklet H. V_H is defined as $V_H = \left\{v_{h_p} | p = 1, 2, \ldots, featureNum\right\}$ which is a set of feature vectors v_{h_p} from the past position h of tracklet H. $featureNum$ denotes the number of face features derived from H, and satisfies $featureNum \leq historyNum$ since a face may not be detected at a past position.

To make the association, the face appearance model needs to calculate the similarity between one detect and one tracklet—that is, the distance between v_d and V_H. We propose three distance measurements to consider various environments when calculating the distance.

The first distance measurement uses the minimum distance between v_d and V_H as follows:

$$l_{min} = \underset{p}{min}(\left\|v_d - v_{p_h}\right\|_2). \tag{11}$$

The second distance measurement uses the average distance between v_d and V_H as follows:

$$l_{mean} = \frac{1}{featureNum} \sum_{h=1}^{featureNum} (\left\|v_d - v_{p_h}\right\|_2). \tag{12}$$

The third distance measurement uses the distance between v_d and the most recent vector $v_{p_{featureNum}}$ of V_H as follows:

$$l_{last} = \left\|v_d - v_{p_{featureNum}}\right\|_2. \tag{13}$$

The distance between the detect and the tracklet is measured by selecting the appropriate measurement from the three proposed distance measurements. If the distance is less than the threshold τ^{face}, it is determined to be the same person and the association can be proceeded. The detailed association procedure is defined with the following Algorithm 1.

Algorithm 1: Proposed face appearance model algorithm

```
Input   : $X = \{x_1, x_2, \ldots, x_{end}\}$   # Image sequence
          $\Delta = \{D_1, D_2, \ldots, D_{end}\}$   # Detection result set
Output: $\{T_1, T_2, \ldots, T_{end}\}$   # Tracking result set
Data    : $D = \{d_1, d_2, \ldots, d_{detectsNum}\}$   # Detection set for a frame
          $T = \{H_1, H_2, \ldots, H_{tracksNum}\}$   # Track set for a frame
          $H = \left\{h_1, h_2, \ldots, h_{historysNum}\right\}$   # The past position set of an object being
tracked
          $d$   # Location of a detected object
          $h$   # The past position of an object being tracked
Initialization: $T \leftarrow \emptyset$
foreach $x, D \in (X, \Delta)$ do
    $T^{match}, T^{lost}, D^{match}, D^{candidate} \leftarrow BaselineTracker.AssociationMethod(x, D)$
    foreach $H \in T^{lost}$ do
        $V_H \leftarrow FaceEmbeddingModule(FaceDetector(H))$
        foreach $d \in D^{candidate}$ do
            $v_d \leftarrow FaceEmbeddingModule(FaceDetector(d))$
            $l \leftarrow DistanceCalculate(V_H, v_d)$   # by Equations (11)–(13)
            if $l < \tau^{face}$ then
                $T^{match} \leftarrow T^{match} \cup H,\ T^{lost} \leftarrow T^{lost} - H$
                $D^{match} \leftarrow D^{match} \cup d,\ D^{candidate} \leftarrow D^{candidate} - d$
                $Association(H, d)$
                $break$
            end
        end
    end
    $BaselineTracker.NewTracks(D^{candidate})$
end
```

4. Experiment

4.1. Experiment Configuration

Baseline tracker. Two baseline trackers are selected according to several conditions to test our method. The first condition is that our method aims to overcome the problems occurring in the real-time environment; thus, we apply the proposed method to the online MOT framework. The second is to select validated trackers that are listed in the MOT Challenge [22]. The last condition is that open-source project models are selected to avoid polarization and to evaluate the performance fairly. In accordance with those conditions, we adopt MOTDT [28] and DeepSORT [27] as baseline trackers. MOTDT proposes a Faster RCNN-based hitmap generation model to filter the correct candidates from the expanded candidates by combining the current detection results with the previous tracking results. This approach shows effectiveness in situations where the data is noisy because the detector additionally finds objects that failed detection. DeepSORT uses the Kalman filter-based motion model proposed by SORT [26] and proposes a CNN-based appearance model. DeepSORT has a high dependency on detection results when constructing candidates, but it is effective in dynamic situations due to the simple association method. For the human detector to provide detection results for the two baseline trackers, we use YOLOv2 [23] 544 × 544 trained with the VOC (2007 + 2012) [41] dataset.

Restoration module. We use the SRGAN (super-resolution GAN) [36] as the image restoration module used in the re-detection and site restoration sections. SRGAN applies an adversarial loss to solve the super resolution problem and proves that the pixel values to be interpolated can be located in a realistic manifold. It is more effective than existing models using MSE (mean squared error). Based on these properties, SRGAN is used to estimate high-resolution sharp images from low-resolution blurry images. The traditional SRGAN uses the high-resolution image and the reduced low-resolution image for the ground truths and the input data, respectively, to learn to estimate the high-resolution image from the low-resolution image. However, in order for the tracker to adapt properly to dynamic movements, it is necessary not only to estimate high-resolution images from low-resolution images but also to estimate sharp images from blurry images. Therefore, our image restoration module is trained using high-resolution sharp images for the ground truths and low-resolution blurry images for the input data. Figure 5 shows how to construct a dataset to train our image restoration module.

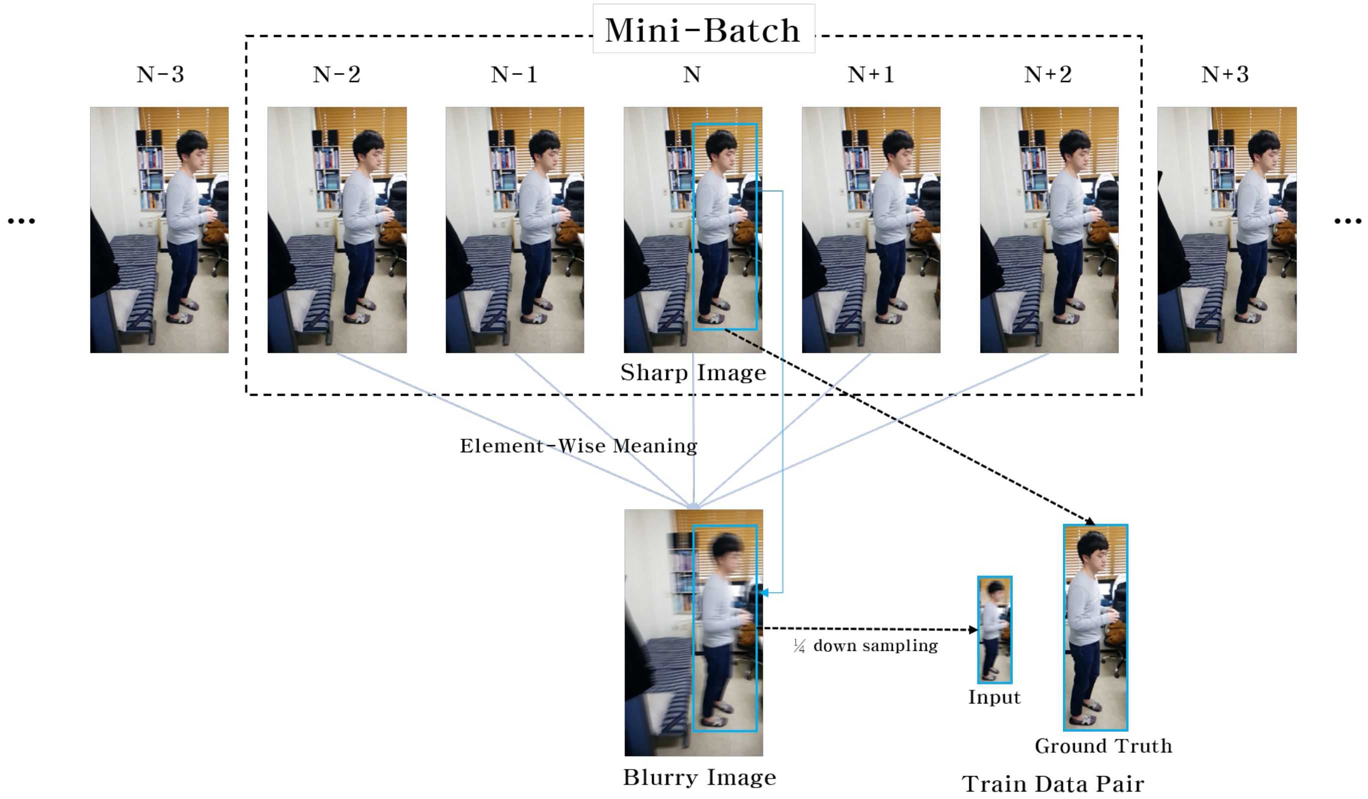


Figure 5. The construction of a training data pair for the image restoration module.

To create one training data pair consisting of a low-resolution blurry image and a high-resolution sharp image, one chunk needs to be configured. This chunk is constructed by selecting an odd number of images from a set of images arranged in a chronological order. From the constructed chunk, the low-resolution blurry image can be obtained by averaging the images and scaling it down to a quarter, and high-resolution sharp images can be obtained from the middle image of the chunk. In order for our image restoration module to focus on restoring a person's image, we need to exclude the background of the learning image. Therefore, we detect and crop humans in high-resolution sharp images and crop the same positions in low-resolution blurry images to produce training data pairs.

There are a few conditions in taking a video to construct a training dataset:

1. It should aim for people who are not included in the benchmark set for fair evaluation.
2. To simulate a natural and precise blurry image, the image must be taken at a high refresh rate with dynamic movement.

According to the above conditions, images are taken at a 240 Hz refresh rate, the chunk size is set to 7, and datasets of 31,640 pairs (body 21,220 pairs, face 10,420 pairs) are extracted from the images. The configuration for learning is based on the SRGAN default setting with 400 epochs. Figure 6 shows the examples of the restoration using our image restoration module.

Face appearance model. For the face detector, we use a mobileNet SSD [25] trained with a WIDERFACE dataset, which follows the mobileNet default configuration of the object detection API provided by the TensorFlow official project. FaceNet [39] is used for the face embedding, which is trained to represent face features in 128d using triplet loss with the MS-Celeb-1M dataset. The input image size is 224 × 224 according to the FaceNet default setting. When resizing, the interpolation method uses the inter-linear method.

The reconstructed images from the image restoration module may have a positive effect on the face appearance model. However, even if those images have visually realistic results, they may adversely affect the recognition performance. We thus evaluate the improvement in the performance of the face recognition compared to the existing interpolation method when ×4 up-sampling low-resolution face images are restored using our image restoration module. To measure the performance of the face recognition of both methods, we use the benchmarking method using the LFW(Labeled Faces in the Wild) dataset [42] proposed by FaceNet.

Figure 7 shows the re-id benchmark results. The performance with the size of 20 × 20 is lower than that of bicubic interpolation because the GAN model generates some artifacts due to the severely lacking information, but for 30 × 30 to 60 × 60, superior results are shown when using our method. Since the amount of information for the recognition is large enough for over 70 × 70, the difference in performance is insignificant. Consequently, our restoration module positively affects the appearance model.

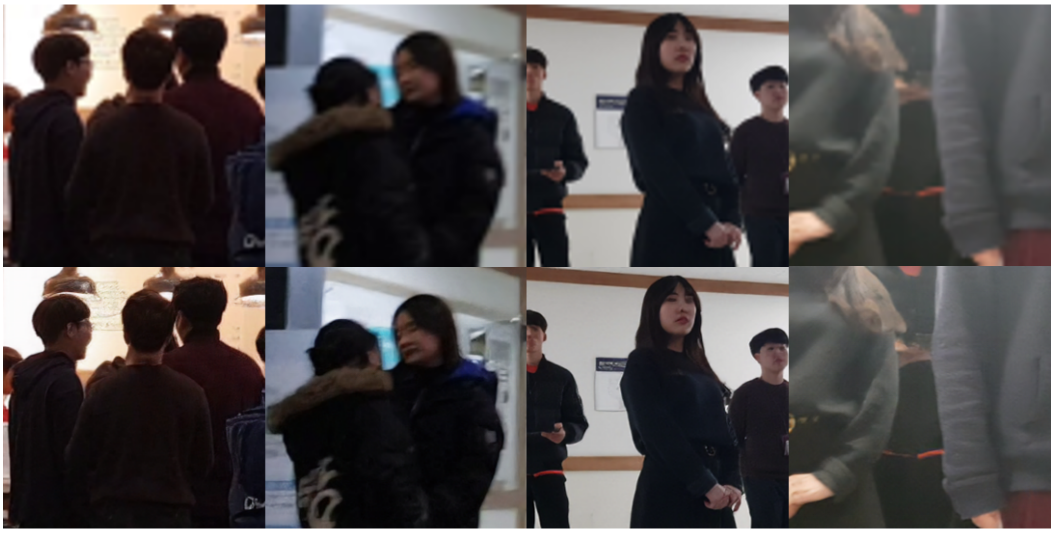


Figure 6. The examples of the image restoration. The upper and lower parts represent the raw input images and the inferred images, respectively.

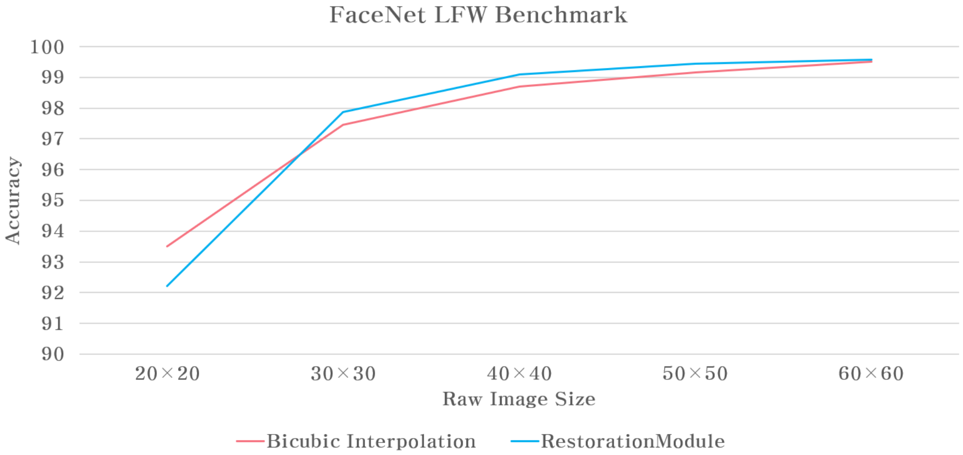


Figure 7. The comparison of the identification performances of the restored images using the bicubic interpolation method and the image restoration module. The FaceNet default settings are used to extract the identification performance. The vertical axis represents the accuracies and the horizontal axis is the size of the original image to be up-sampled.

4.2. Benchmark Set

Our methods are proposed to enhance the performance by overcoming problems in dynamic situations. To see the contributions of our methods, we construct a robot environment dataset and evaluate the performance according to the MOT16 [22] benchmark method. A Turtlebot v2 is used for mobility and the recording camera is a Galaxy S8 + 12MP mounted on the robot. The video is taken following the scenario where the robot interacts with the user or patrols inside the building, and the ground truths are made by labeling humans' bounding boxes with a handcraft according to the manner of the construction of the MOT16 benchmark set. Table 1 lists the details of our benchmark sets.

Table 1. The details of benchmark sets built on three different scenarios.

	Interaction-1	**Interaction-2**	**Patrol**
Frame	2700	1242	2868
Bounding box	5814	6201	9913
ID	38	68	133
Explanation	destination guidance 1	destination guidance 2	patrol
FPS	30	30	30
Image size	1920 × 1080	1920 × 1080	1920 × 1080
Length(sec)	90	41.4	95.6
Crowd	low	middle	high

The Interaction-1 and Interaction-2 benchmark sets are based on the robotic guidance scenarios. The guidance robot interacts with the person and moves to its service destination. During the process, the users follow the robot with nonlinear movements that are difficult to predict. Interaction-1 and Interaction-2 include a small number of people and a relatively large number of people, respectively.

The Patrol benchmark set is based on a robotic patrol scenario. If the robot has no current purpose in progress, it patrols the lobby until a new command is received from the user. A large number of people appear in the scenes, and the robot repeats the actions of getting close to and away from the people to patrol.

Our dataset contains more dynamic scenes than the MOT16 dataset because it is based on images taken from a robot in service. To validate the degree of dynamic movements, we quantify them with Laplacian variance, taking advantage of the fact that a dynamic movement inevitably produces motion blur. Figure 8 represents the Laplacian variances in our dataset in comparison with the MOT16 datasets.

All three datasets used in our experiments are found to show lower Laplacian variances compared to the MOT16 datasets. This indicates that the edge detection by the Laplacian kernel is difficult due to the dynamic movements.

We are concerned about whether the reason for the low Laplacian variances is the camera characteristics or not. To address this concern, we additionally constructed an image set, Robot-Stop, which observes moving people in the stationary condition of the robot. The experimental results demonstrate that the low Laplacian variances of our datasets are not given by the camera characteristics because the second highest result was observed amongst all the datasets.

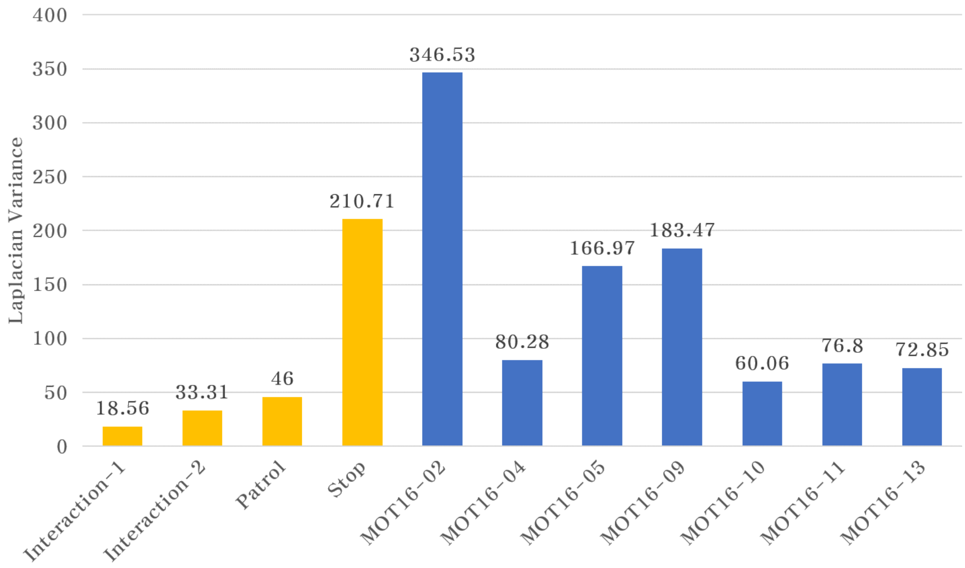


Figure 8. Measured Laplacian variances of each dataset. The horizontal and vertical axis represents the dataset and the averaged value of the Laplacian variance for each dataset, respectively.

4.3. Experiment Results

Evaluation metric. We use Multiple Object Tracking Accuracy (MOTA) [43], ID F1 score [44], the ratio of Mostly Tracked targets (MT), the ratio of Mostly Lost targets (ML), and the number of ID Switches (IDS) as the performance metrics, which are significant among several metrics used in the MOT Challenge. Specifically, MOTA and IDF1 are considered to be the most important performance metrics. MOTA represents false positives, missed targets, and identity switches together, and IDF1 represents the consistent tracking rate of object ID. In our experiments, we consider the MOTA score as the first priority and the IDF1 score as the second priority in performance. For the evaluation, the MATLAB-based MOT Challenge Development Kit is used according to the MOT16 benchmark rule, and the three dataset benchmarking results are given with their weighted average scores.

Evaluation baseline tracker. The default τ^{detect} value of the detector may provide biased performance to some trackers. Therefore, to avoid this, we first observe the performances of the selected detector's detect confidence threshold, τ^{detect}, and fix the threshold that results in the highest performance for the baseline tracker.

In the experimental results for the baseline shown in Figure 9, the best values of τ^{detect} are 0.31 for MOTDT and 0.40 for DeepSORT. We specify MOTA and IDF1 scores for the thresholds as the baseline performance to compare with our methods.

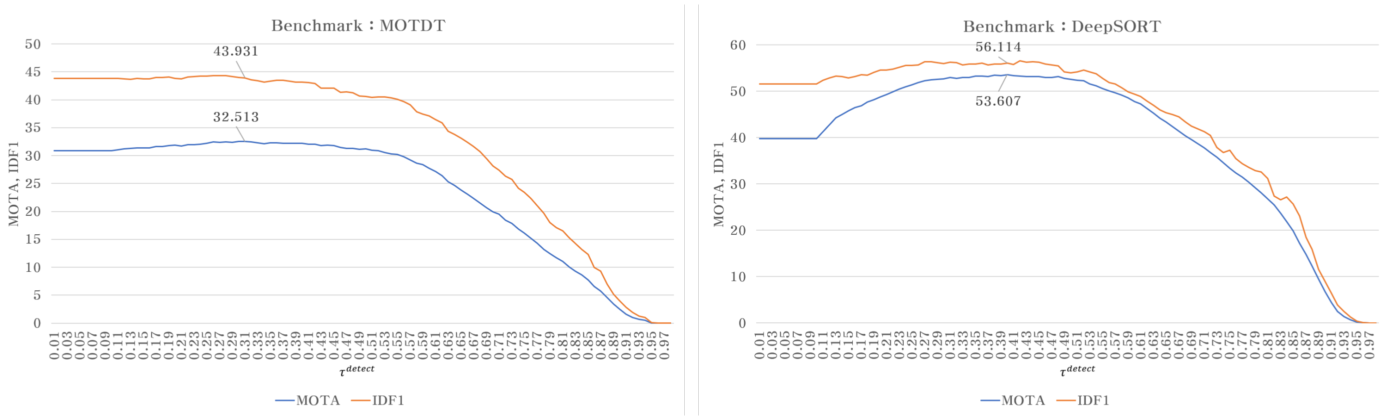


Figure 9. MOTA and IDF1 scores varying τ^{detect} for two baseline trackers, MOTDT and DeepSORT.

Evaluation and ablation studies. The finalized model proposed in this paper is combined with three aforementioned methods. The ablation study confirms the effectiveness of the combination of each method. When combining methods, we set parameters that record the highest performance of each of the three methods without the additional hyperparameter settings for each combination.

Table 2 shows the experimental results of the ablation study. The arrows after evaluation metrics indicate that the higher (↑) and lower (↓) values represent the better performance. In this table, specifically, the method column refers to which of the three methods—the re-detection, the site restoration, and the face appearance model (simplified as "face appearance")—are applied to the corresponding baseline tracker.

Table 2. Experimental results of ablation studies based on the baseline trackers, MOTDT and DeepSORT. Highlighting represents the best performance.

Baseline	Method	MOTA (↑)	IDF1 (↑)	MT (↑)	ML (↓)	ID sw (↓)
MOTDT	-	32.513	43.931	40	120	162
	re-detection	32.641	44.017	41	118	172
	site restoration	32.818	44.263	41	120	163
	face appearance	32.832	44.226	42	120	**159**
	re-detection site restoration	32.959	44.355	42	118	173
	re-detection face appearance	32.886	44.492	43	118	163
	site restoration face appearance	33.118	44.419	43	120	160
	re-detection site restoration face appearance	**33.164**	**44.531**	**44**	**118**	167
DeepSORT	-	53.607	56.114	83	69	132
	re-detection	53.929	57.616	84	67	137
	site restoration	53.707	56.165	84	69	130
	face appearance	53.834	56.471	83	68	123
	re-detection site restoration	54.007	57.656	85	67	137
	re-detection face appearance	54.093	**57.943**	85	67	134
	site restoration face appearance	53.989	56.709	84	68	**117**
	re-detection site restoration face appearance	**54.180**	57.906	**85**	**67**	130

The proposed models generally perform better than the baseline methods, even when using some of our methods, and the best when all methods are combined. Specifically, comparing our finalized model with the baseline, the MOTA score is improved by 0.65% in MOTDT, 0.57% in DeepSORT, and the IDF1 score by 0.6% in MOTDT and 1.79% in DeepSORT.

Analysis of re-detection. Two experiments are conducted to adjust the aforementioned thresholds, τ^{amb} and $\tau^{redetect}$, to find the optimal parameters for the re-detection method. In the first experiment, τ^{amb} ranges from 0 to τ^{detect}, and $\tau^{redetect}$ ranges from τ^{detect}

to 0.9. Those parameters are adjusted in units of 0.05 to identify tendencies in performance. Delicate evaluation is conducted in the second experiment, adjusting those parameters to 0.01 units for the ±0.05 range on the value which shows the maximum MOTA score in the first experiment and where the IDF1 score was above the baseline. Figures 10 and 11 represent the results of the first and second experiments, respectively, where the green dotted line represents the baseline performance.

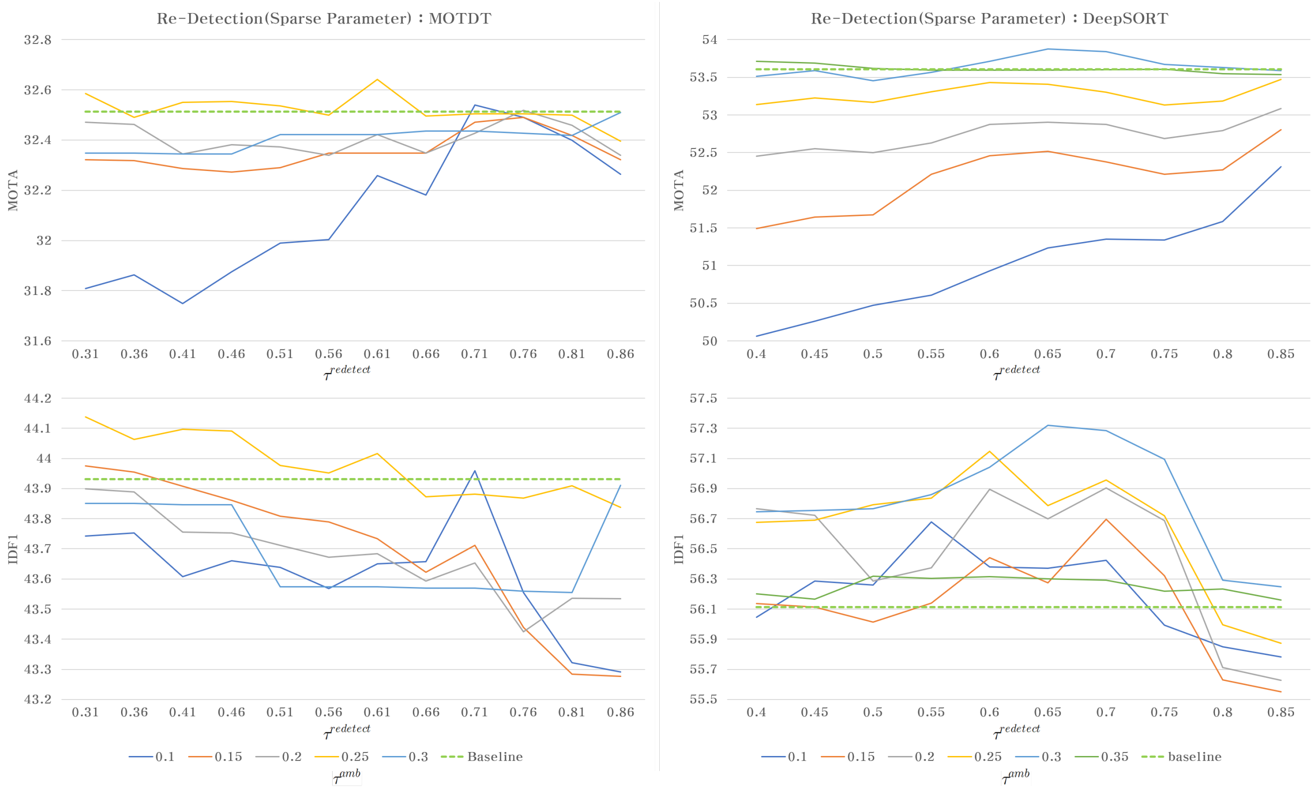
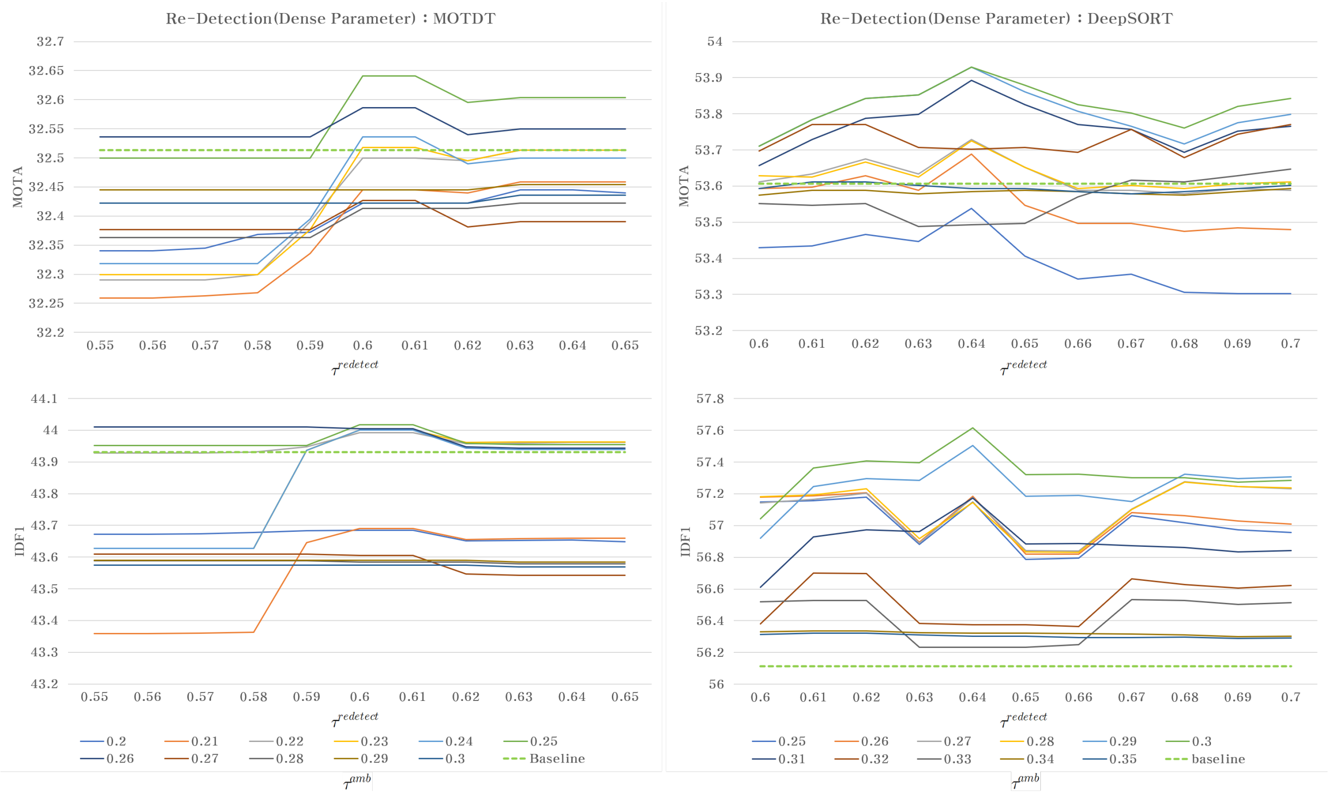


Figure 10. The results of the first experiment varying $\tau^{redetect}$ and τ^{amb} in 0.05 units. The vertical and horizontal axes represent the MOTA and IDF1 score and $\tau^{redetect}$, respectively. Specifically, the green dotted line and the multiple colors of other lines represent the baseline performance and each τ^{amb}, respectively.

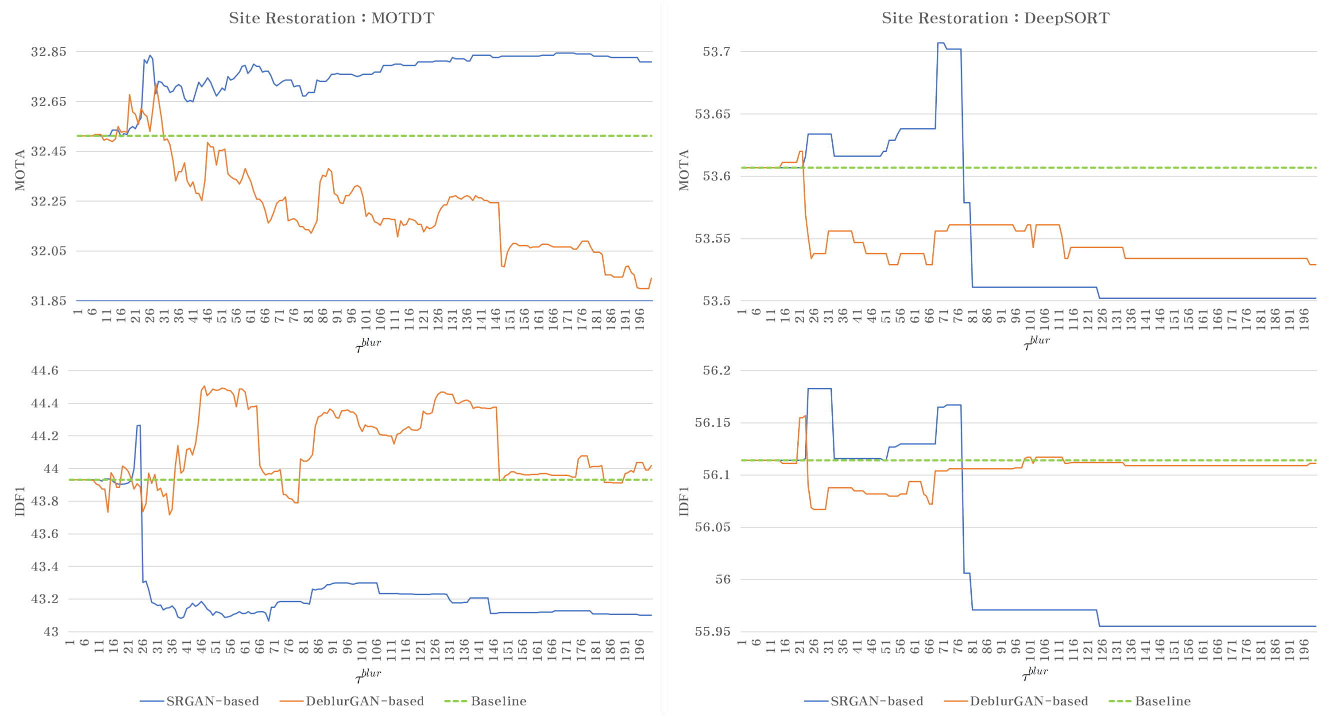
According to the results shown in Figure 11, the highest score is achieved when the value of τ^{amb} and $\tau^{redetect}$ are 0.25 and 0.61 in MOTDT and the value of τ^{amb} and $\tau^{redetect}$ are 0.3 and 0.64 in DeepSORT, respectively. The reason why DeepSORT is enhanced more than MOTDT when using re-detection is that DeepSORT uses only detect results as tracking linkage candidates. This indicates that DeepSORT is more dependent on detect results. On the contrary, MOTDT uses not only the detect results but the locations estimated by the motion model as an association candidate. Consequently, DeepSORT is more affected in the performance improvement by our re-detection method.

Analysis of site restoration. We perform an experiment to adjust the aforementioned threshold, τ^{blur}, to find the optimal parameter for the best configuration of the site restoration method. As an additional experiment, we compare SRGAN, which we use in the image restoration module, with DeblurGAN, a representative model based on GAN for the Deblur problem. DeblurGAN is trained on our the same dataset as SRGAN, and the detailed training configuration follows the default settings suggested in the paper. Figure 12 shows the effect of the image restoration module on the appearance model by adjusting the τ^{blur} by 1 unit for the range from 0 to 200.


Re-Detection(Dense Parameter) : MOTDT

Re-Detection(Dense Parameter) : DeepSORT

Figure 11. The experimental results of the second experiment to find delicate parameters based on the parameters that performed the best in the first experiment. $\tau^{redetect}$ and τ^{amb} vary in units of 0.01 ranged over the value of the base parameter.


Site Restoration : MOTDT

Site Restoration : DeepSORT

Figure 12. Experimental results to evaluate the effect of SRGAN- or DeblurGAN-based image restoration modules on the appearance model of the baseline tracker. The experiment proceeds with increasing τ^{blur} by 1 unit.

As illustrated in Figure 12, the results show the highest performance for τ^{blur} with the value of 24 on MOTDT and for τ^{blur} with 70 on DeepSORT. They indicate that MOTDT is effective in severely dynamic situations, but DeepSORT is effective in relatively weakly dynamic situations. The further experimental results represent the predominance of SRGAN in most cases. In fact, the images restored by DeblurGAN tend to make blur effects disappear more clearly. However, when we zoomed in on the image as shown in Figure 13, specific patterns that are newly created are observed. We speculate that this pattern has a negative effect on the appearance model.

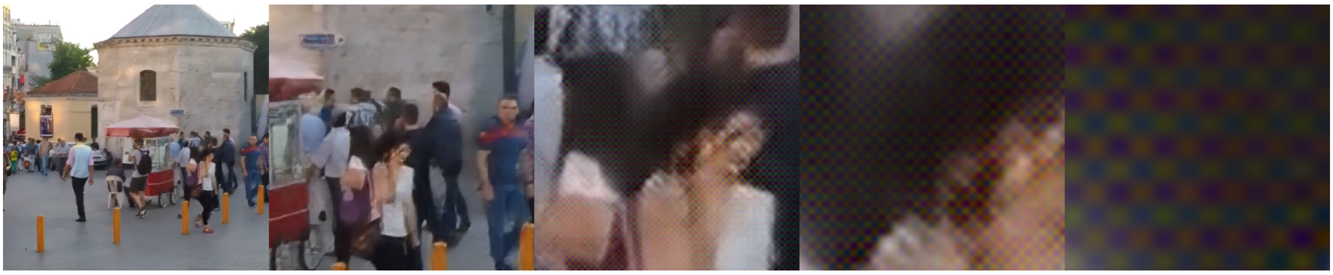


Figure 13. An inferred image by DeblurGAN. We apply the learning method and dataset proposed in [37].

Analysis of the face appearance model. We conduct an experiment with three distance measurements (Equations (11)–(13)) to find an optimal parameter for τ^{face} in the face appearance model. Figure 14 shows the effect of the face appearance model on the baseline tracker adjusting τ^{face} by 0.01 unit for the range from 0 to 1 for each distance measurement.

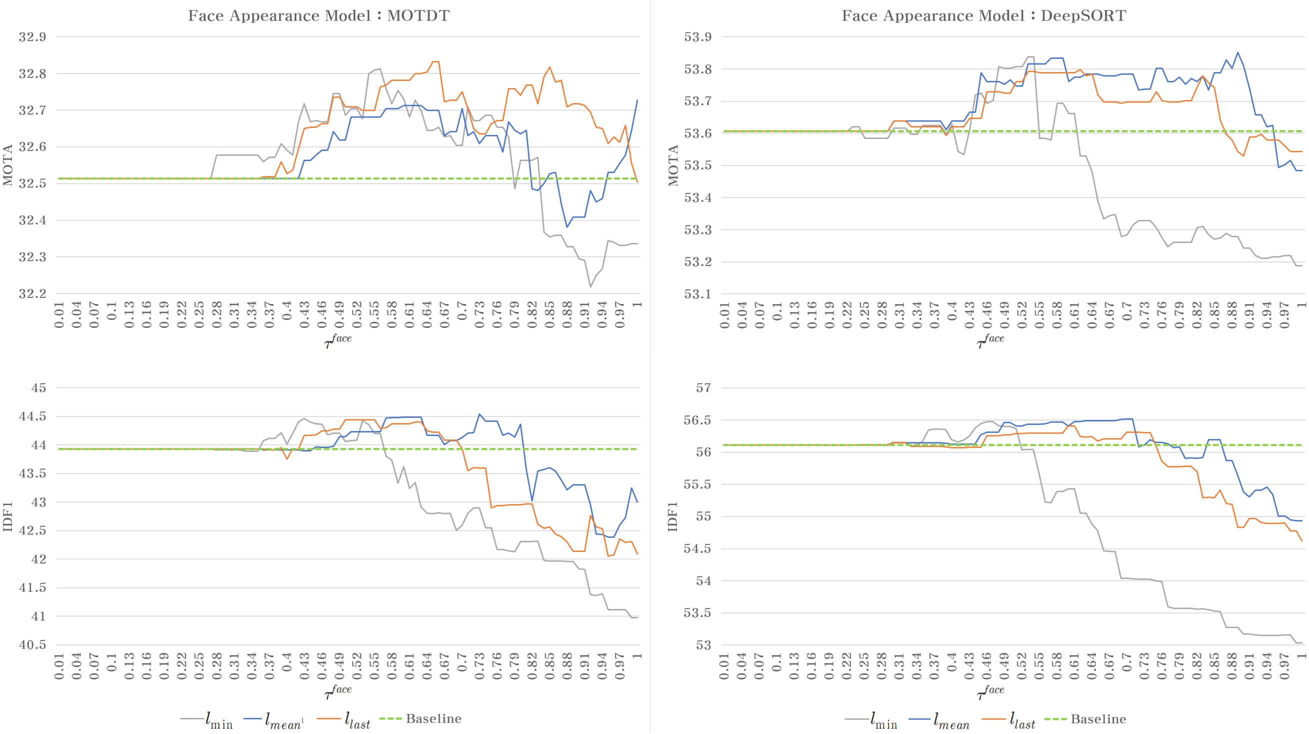


Figure 14. Experimental results to validate our proposed face appearance model. Specifically, the color of each line represents the corresponding distance measurement.

As shown in Figure 14, MOTDT achieves the highest performance when its distance measurement is l_{last} and the value of τ^{face} is 0.65. This indicates that, because MOTDT has a relatively large number of tracking candidates, using the most recent face information contributes to the better performance rather than using all of the past face information. Moreover, when only the face information of the last state is used, the embedding vector distances are relatively close due to the rare changes of the face. The performance is thus improved at a relatively strong threshold. DeepSORT achieves the highest performance when the distance measurement is l_{mean} and the value of τ^{face} is 0.58. Because DeepSORT relies on the detection results to configure tracking candidates, comparing the limited candidates with the past face information more intensively contributes to the better performance. In addition, since the face information of the entire past states is used, various embedding vectors should be considered, resulting in the improved performance at relatively weak thresholds.

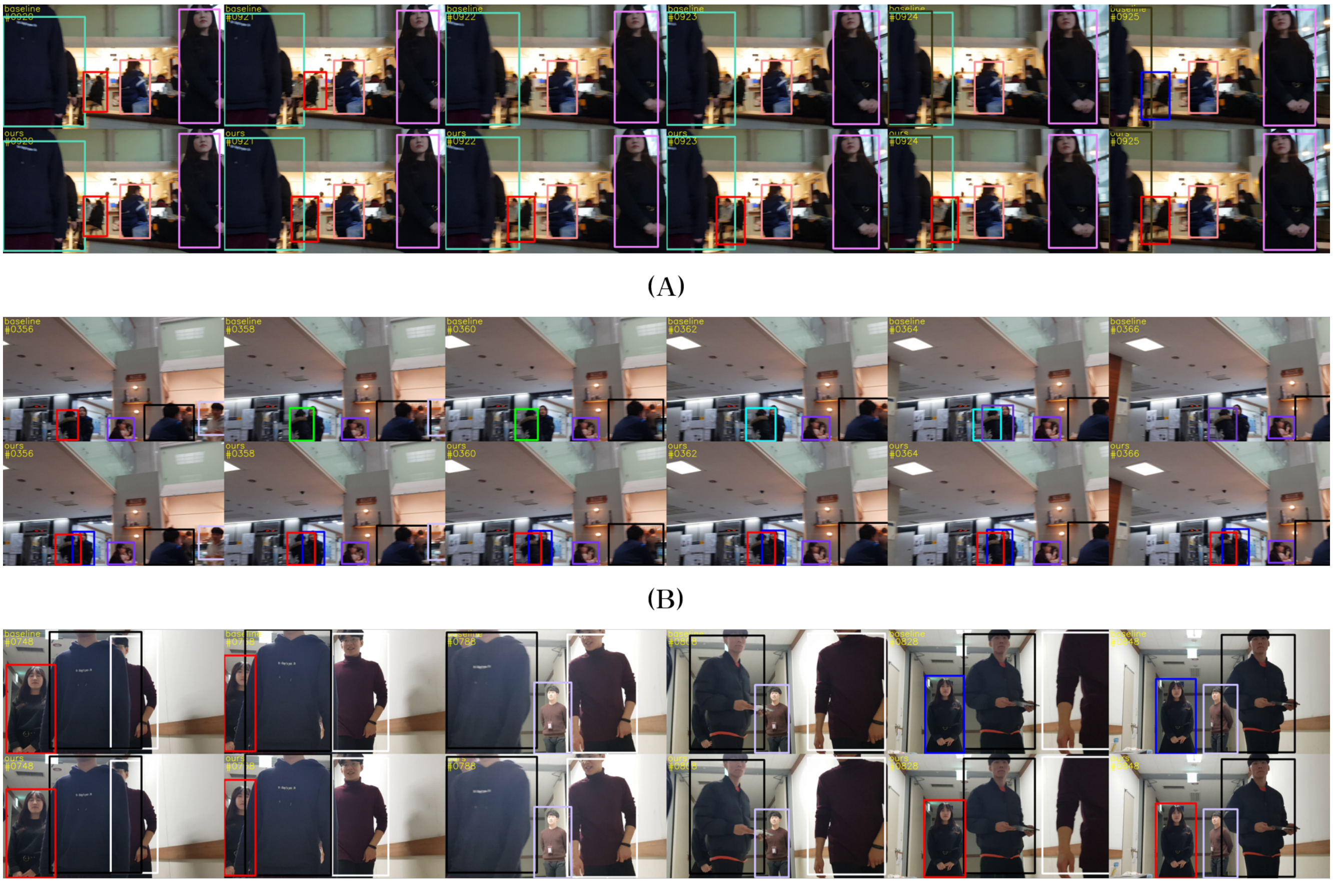
Experiments on MOT challenge datasets. In order to check how effective the finished model is when applied to an unfamiliar environment, experiments are conducted in a new environment using parameters found through previous experiments. We use the MOT Challenge dataset for simulating a new environment. Among all MOT Challenge datasets, only four of those which are based on the moving platform (MOT16-(05,10,11,13)) are used to match the scope of our paper. The experimental results are reported in Table 3, and the arrows after evaluation metrics indicate that the higher (↑) and lower (↓) values represent the better performance.

Table 3. Experimental results to confirm the effect of the proposed model applied to an unfamiliar environment. The parameters found through the previous experiment are used without change, and the MOT Challenge dataset is used as a test set. Highlighting represents better performance.

Baseline	Test Data	Method	MOTA (↑)	IDF1 (↑)	MT (↑)	ML (↓)	ID sw (↓)
MOTDT	MOT16-05	-	36.037	**47.269**	9	51	**53**
		ours	**36.066**	46.419	9	51	57
	MOT16-10	-	13.492	22.463	3	40	**46**
		ours	**13.785**	**23.129**	**4**	**38**	48
	MOT16-11	-	39.481	42.308	5	39	**31**
		ours	**39.863**	**42.91**	5	**38**	34
	MOT16-13	-	3.7642	11.057	0	94	14
		ours	**3.8253**	**11.188**	0	**93**	**13**
DeepSORT	MOT16-05	-	**38.09**	45.215	13	53	41
		ours	37.87	**45.293**	13	53	41
	MOT16-10	-	7.9477	17.072	1	40	**34**
		ours	**8.1507**	**18.532**	1	40	35
	MOT16-11	-	34.794	32.311	7	39	67
		ours	**34.848**	**32.968**	**8**	39	**66**
	MOT16-13	-	2.5328	7.8663	0	97	8
		ours	**2.5764**	**8.3659**	0	97	**7**

The experimental results show the performance improvement of the main metrics, MOTA and IDF1, in the datasets (except for one of the four datasets). The reason why performance has not improved in some datasets can be interpreted as a problem of excessively low image quality. A dataset with improved performance (MOT16-(10,11,13)) has a high resolution of 1920 × 1080 and has little noise, whereas a dataset without improved performance (MOT16-05) has a low resolution of 640 × 480 and a lot of noise. Since low resolution and a lot of noise cause reasoning failure of the Image Restoration Module and Face Embedding Module, the threshold value used for verification needs to be strict. Therefore, when the image quality is significantly different from the experimental environment, retuning the $\tau^{redetect}$, τ^{blue}, and τ^{face} values to strict values may be advantageous for improving performance. From the overall experimental result, it is validated that our proposed methodology enhances the tracking performance when applied to a general situation.

Qualitative results. Several qualitative assessments are conducted, summarizing some of the benchmark results. DeepSORT is selected for evaluation as the baseline tracker, and Figure 15 represents three qualitative evaluation results. The top row of each assessment is the benchmark result of the baseline tracker, and the bottom one is that of our proposed model on the baseline tracker. The bounding boxes with the same color in each evaluation result are meant to be recognized as the same objects by the tracker. In particular, the red bounding boxes represent the object which shows the largest difference between the proposed model and the baseline tracker. The examples clearly show that the three methods we propose are capable of maintaining the IDs of objects properly by overcoming the visual changes and the congestion of objects that occur in dynamic situations.



(A)

(B)

(C)

Figure 15. Qualitative examples of our finalized model with all three proposed methods based on the DeepSORT baseline. (**A**), (**B**), and (**C**) denote the experimental results in the Patrol, Interaction-1, and Interaction-2 datasets, respectively.

(A) shows the selected frames in the Patrol dataset. This dataset causes severe motion blur, especially for distant objects due to the camera rotation. In the case of the baseline, the detection of the object fails, which is marked in red, and the tracking is terminated. On the other hand, the ID is constantly tracked in red on our model because it maintains the detection in a high success rate, even for blurry objects, by performing the re-detection based on the restoration functionality.

(B) shows an example of the experimental results in the Interaction-2 dataset that has a lot of occlusions between objects that are relatively far apart. In the case of the baseline, the association failure of the object indicated by the red bounding box occurs due to the noise generated by the dynamic movement. On the contrary, our method maintains the object's ID constantly because the accuracy of the appearance model is improved by restoring the noise using the site restoration method.

(C) includes the experimental examples on the Interaction-1 dataset in which humans are observed at close range and many occlusions are detected due to movements. In particular, for the occlusion of the objects indicated by the red and black bounding boxes, the tracking is terminated after the occlusion and a new ID is assigned in the case of the baseline. The reason for the mistracking is the confusion of the appearance model caused by the mixed features. On the other hand, in our method, the ID remains intact because the use of a face feature alleviates mixing of features.

5. Conclusions

In this paper, we propose three methods to enhance the performance of multiple object tracking, especially in a dynamic environment. First, re-detection and site restoration methods use the approach to remove noises from an image to improve the detection and identification performance, respectively. To remove the noises of the image, we implement the image restoration module by adopting and learning the GAN-based image inference model suitable for the dynamic environment. Moreover, the face appearance model uses an approach that uses face features to reduce the likelihood of an association failure. We design three distance measurements to efficiently calculate the distance between multiple features so that our appearance model can be applied to a general-purpose environment. In order to validate the effectiveness of our proposed methods, we construct dynamic robot environments and conduct experiments with robot service scenarios. As a result, the performance of the multiple object tracking is improved significantly due to the adaptability of our proposed model to the dynamic environment in comparison with the existing trackers. The image restoration module proposed by us has a limitation in that it cannot utilize the characteristics of time series data because it restores using only one image. In the future, if a recurrent architecture-based image inference model using a neural network is applied to an existing image restoration module, better performance is expected to be achieved for dynamic situations using the time series characteristics of the dataset as well.

Author Contributions: Conceptualization, J.S.K.; methodology, J.S.K. and D.S.C.; software, J.S.K.; validation, J.S.K. and D.S.C.; formal analysis, J.S.K.; investigation, J.S.K. and Y.S.C.; resources, J.S.K. and D.S.C.; data curation, J.S.K.; writing—original draft preparation, J.S.K.; writing—review and editing, J.S.K., D.S.C., and Y.S.C.; visualization, J.S.K.; supervision, Y.S.C.; funding acquisition, Y.S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2020-0-01373, Artificial Intelligence Graduate School Program (Hanyang University)), the Technology Innovation Program (10077553, Development of Social Robot Intelligence for Social Human-Robot Interaction of Service Robots) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea), and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1A2C1014037).

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

References

1. Gu, R.; Wang, G.; Hwang, J.N. Efficient multi-person hierarchical 3D pose estimation for autonomous driving. In Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 28–30 March 2019; pp. 163–168.
2. Hsu, H.M.; Huang, T.W.; Wang, G.; Cai, J.; Lei, Z.; Hwang, J.N. Multi-Camera Tracking of Vehicles based on Deep Features Re-ID and Trajectory-Based Camera Link Models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 416–424.
3. Tang, Z.; Naphade, M.; Liu, M.Y.; Yang, X.; Birchfield, S.; Wang, S.; Kumar, R.; Anastasiu, D.; Hwang, J.N. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 8797–8806.

4. Tang, Z.; Wang, G.; Xiao, H.; Zheng, A.; Hwang, J.N. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 108–115.
5. Wang, G.; Yuan, X.; Zheng, A.; Hsu, H.M.; Hwang, J.N. Anomaly Candidate Identification and Starting Time Estimation of Vehicles from Traffic Videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 382–390.
6. Tian, W.; Lauer, M.; Chen, L. Online multi-object tracking using joint domain information in traffic scenarios. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 374–384.
7. Wang, X.; Fan, B.; Chang, S.; Wang, Z.; Liu, X.; Tao, D.; Huang, T.S. Greedy batch-based minimum-cost flows for tracking multiple objects. *IEEE Trans. Image Process.* **2017**, *26*, 4765–4776. [CrossRef]
8. Keuper, M.; Tang, S.; Zhongjie, Y.; Andres, B.; Brox, T.; Schiele, B. A multi-Cut Formulation for Joint Segmentation and Tracking of Multiple Objects. *arXiv* **2016**, arXiv:1607.06317.
9. Yang, B.; Nevatia, R. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 1918–1925.
10. Wang, B.; Wang, G.; Luk Chan, K.; Wang, L. Tracklet association with online target-specific metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 1234–1241.
11. Wei, J.; Yang, M.; Liu, F. Learning spatio-temporal information for multi-object tracking. *IEEE Access* **2017**, *5*, 3869–3877. [CrossRef]
12. Milan, A.; Rezatofighi, S.H.; Dick, A.; Reid, I.; Schindler, K. Online multi-target tracking using recurrent neural networks. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
13. Fu, Z.; Feng, P.; Angelini, F.; Chambers, J.; Naqvi, S.M. Particle PHD filter based multiple human tracking using online group-structured dictionary learning. *IEEE Access* **2018**, *6*, 14764–14778. [CrossRef]
14. Chu, Q.; Ouyang, W.; Li, H.; Wang, X.; Liu, B.; Yu, N. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4836–4845.
15. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577. [CrossRef]
16. Pinho, R.R.; Tavares, J.M.R.; Correia, M.V. A movement tracking management model with Kalman filtering, global optimization techniques and mahalanobis distance. *Adv. Comput. Methods Sci. Eng.* **2005**, *4*, 1–3.
17. Pérez, P.; Hue, C.; Vermaak, J.; Gangnet, M. Color-based probabilistic tracking. In *European Conference on Computer Vision (ECCV)*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 661–675.
18. Li, Y.; Ai, H.; Yamashita, T.; Lao, S.; Kawade, M. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1728–1740. [PubMed]
19. Dai, S.; Wu, Y. Motion from blur. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, USA, 24–26 June 2008; pp. 1–8.
20. Cho, S.; Matsushita, Y.; Lee, S. Removing non-uniform motion blur from images. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, 14–20 October 2007; pp. 1–8.
21. Potmesil, M.; Chakravarty, I. Modeling motion blur in computer-generated images. *ACM Siggraph Comput. Graph.* **1983**, *17*, 389–399. [CrossRef]
22. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; pp. 779–788.
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
25. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
26. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
27. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
28. Chen, L.; Ai, H.; Zhuang, Z.; Shang, C. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
29. Ning, G.; Zhang, Z.; Huang, C.; Ren, X.; Wang, H.; Cai, C.; He, Z. Spatially supervised recurrent convolutional neural networks for visual object tracking. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4.

30. Chen, L.; Ai, H.; Shang, C.; Zhuang, Z.; Bai, B. Online multi-object tracking with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 645–649.
31. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 June 2014; pp. 2672–2680.
32. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
33. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Puerto Rico, USA, 24–30 June 2017; pp. 1125–1134.
34. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning(ICML), Sydney, Australia, 6–11 August 2017; pp. 1857–1865.
35. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2223–2232.
36. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Puerto Rico, USA, 24–30 June 2017; pp. 4681–4690.
37. Kupyn, O.; Budzan, V.; Mykhailych, M.; Mishkin, D.; Matas, J. Deblurgan: Blind motion deblurring using conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8183–8192.
38. Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; Zheng, N. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; pp. 1335–1344.
39. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
40. Pertuz, S.; Puig, D.; Garcia, M.A. Analysis of focus measure operators for shape-from-focus. *Pattern Recognit.* **2013**, *46*, 1415–1432. [CrossRef]
41. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
42. Huang, G.B.; Ramesh, M.; Berg, T.; Learned-Miller, E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*; Technical Report 07-49; University of Massachusetts Amherst: Amherst, MA, USA, 2008.
43. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Eurasip J. Image Video Process.* **2008**, *2008*, 1–10. [CrossRef]
44. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision (ECCV)*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 17–35.