Visual Object Tracking Fusing CNN and Color Histogram based Tracker and Depth Estimation for Automatic Immersive Audio Mixing

Sung-Jun Park[1], Md. Mahbubul Islam[1] and Joong-Hwan Baek[1*]
[1] School of Electronics and Information Engineering
Korea Aerospace University, Goyang 10540, Gyeonggi-do, Korea
[e-mail: tjdwns1011@naver.com, mahbubcse@cu.ac.bd, jhbaek@kau.ac.kr]
*Corresponding Author: Joong-Hwan Baek

Received July 31, 2019; revised November 18, 2019; accepted December 2, 2019; published March 31, 2020

Abstract

We propose a robust visual object tracking algorithm fusing a convolutional neural network tracker trained offline from a large number of video repositories and a color histogram based tracker to track objects for mixing immersive audio. Our algorithm addresses the problem of occlusion and large movements of the CNN based GOTURN generic object tracker. The key idea is the offline training of a binary classifier with the color histogram similarity values estimated via both trackers used in this method to opt appropriate tracker for target tracking and update both trackers with the predicted bounding box position of the target to continue tracking. Furthermore, a histogram similarity constraint is applied before updating the trackers to maximize the tracking accuracy. Finally, we compute the depth(*z*) of the target object by one of the prominent unsupervised monocular depth estimation algorithms to ensure the necessary 3D position of the tracked object to mix the immersive audio into that object. Our proposed algorithm demonstrates about 2% improved accuracy over the outperforming GOTURN algorithm in the existing VOT2014 tracking benchmark. Additionally, our tracker also works well to track multiple objects utilizing the concept of single object tracker but no demonstrations on any MOT benchmark.

***Keywords*:** Immersive Audio, GOTURN, Mean-Shift, CNN, Color Histogram, Depth Estimation

1. Introduction

Tracking objects of interest as an application of immersive audio-based cinema is a cutting edge video capturing technology. In the object-based audio technique, specific soundtracks are mixed with the objects rather than with specific channels using object 3D location, start/end times and other metadata information. The existing audio mixing technologies manually use some input device (e.g. puck) to determine the object 3D position and then mix the corresponding soundtracks. Ambisonics is one of the popular audio applications for handling and delivering full-immersive object-based audio. In our system, we apply two newfangled 2D object tracking algorithm and a depth estimation algorithm to excerpt the trajectory(*XYZ*) of the objects automatically and thereafter the specific soundtracks are assigned throughout the shot or scene of the cinema video.

Visual object tracking is the problem of estimating the trajectory of an object over time by locating its position in every frame of the video. It is considered as one of the fundamental problems in the field of computer vision. The escalation of high specification computers, high-resolution reasonable cameras, and highly dependent video analysis-based applications drive research in object tracking. To date, object tracking is pertinent to the tasks of motion-based recognition, automated surveillance [1], video captioning [2], human-computer interaction [3], traffic monitoring [4], and autonomous vehicles [5, 6].

Although several real-world applications are facilitated by object tracking algorithms there are still many observed difficulties in tracking due to abrupt object motion, changing appearance patterns of both the object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions, and camera motion [7]. For decades, researchers have developed and tried several tracking algorithms in an attempt to solve these different object tracking challenges. For example, Zhang et al. [8], Pan and Hu [9] and Yilmaz et al. [10] proposed algorithms to handle occlusion in the scene. Similarly, Zhong et al. [11], Adam et al. [12], and Babenko et al. [13] proposed algorithms deal with the problem of illumination variations. Despite all the research that has been done to mitigate all of the tracking challenges, not one the tracking algorithms has been adequate enough to meet these challenges.

This paper's scope examines visual object tracking where the object's coordinates are fed into the system as a bounding box at the beginning of the frame sequence and then using tracking algorithms try to automatically track an object in consecutive frames. The workflow of the visual object tracking is shown in **Fig. 1**. Early object tracking algorithms are mainly concentrated on the feature extraction, searching method, and similarity comparison. Statistical techniques [14, 15, 16] and non-statistical techniques [17] are used as a feature extraction method. Particle filter [18] and mean-shift [19, 20] algorithms are usually used for searching the object location. The rapid development of the deep learning networks gives a new dimension in the object tracking research. Automatic feature extraction of the CNN based algorithm shows enviable performance compared to the earlier standard methods. DLT [21] was the first to introduce the deep learning concept in the task of visual object tracking. It performed the training in an offline manner. The combination of generative and discriminative tracking approach makes this method more expressive in terms of image representations than the traditional methods based on principle component analysis (PCA).

These days, mainstreaming of visual object trackers is generally learned online (i.e. during test time) without performing any offline training [17, 13, 22, 23]. The tracking performance of such trackers is not satisfactory due to unexploited the available video

resources. In [17], an online adaptive tracker is proposed that uses a kernelized structured output support vector machine. Kalal et al. [23] proposed an approach that perform long-term tracking of unknown objects into three phases: tracking, learning and detection. They impose a new learning method P-N learning to handle the detector's error if needed. MDNet [24] and C-COT [25] are two CNN-based trackers that are trained online and exhibited, the best performance in VOT2015 and VOT2016 challenges, but the speed is below real-time performance. Another recent CNN-based tracker GOTURN (Generic Object Tracking Using Regression Networks) performs target tracking in real-time through offline training [26]. GOTURN is principally a regression-based approach that only needs a single forward pass through the network to regress the target location without fine-tuning. This network learns from the generic relations of object's appearance and motion through the offline training from huge available data and videos. During tracking, target template and search regions are fed to five individual CNN layers and deep features from two streams are fused into three shared FC layers. These two factors, offline training and single-pass regression accelerate the object tracking speed to 100 fps. Despite GOTURN have achieved enviable performance, the target occlusion missed the target to track because of rapid target interactions. **Fig. 2** is an example of an occluded scene during tracking by GOTURN tracker.

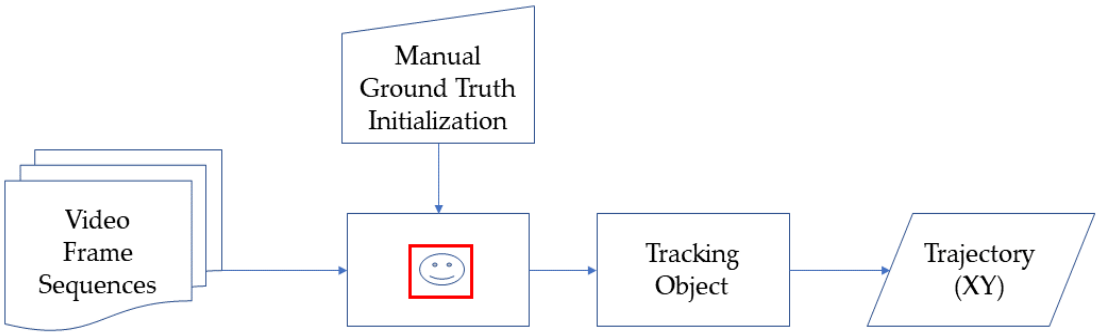


Fig. 1. A procedure flow of detection-free tracking approach

Fig. 2. A scenario of drift case in GOTURN tracker . **Left**: Target object with bounding box (Red) **Middle**: Another object occluded with the target object **Right**: Target changed to another object.

In this paper, we propose a visual object tracking system using a color histogram-based tracker mean-shift is fusing it with GOTURN to improve the overall accuracy of the object tracking system. This successfully provided an improved tracked bounding box position of the target object where GOTURN fails to track the right target. Our algorithm can handle the occlusion and large movement problems that occur in the GOTURN tracker that is depicted in section 4.1. We used Godard et al. [27] algorithm to construct the lost information (depth)

from the 2D image by calculating the range from a projected point to image plane where only a single input image is required without any perception about the geometry of scene or type of objects present. Our proposed algorithm was investigated on the challenging VOT2014 [28] and VOT2015 [29] benchmarks.

Our proposed tracking algorithm also works for multiple target tracking. In recent years, some deep learning-based approaches have been developed in multiple-object tracking [30, 31, 32], but performances are not worth mentioning other than the handcrafted features founded techniques. Constructive thinking leads us to apply deep learning based single object tracker to MOT (Multiple Object Tracking). The qualitative results of multiple object tracking are found in section 4.1.

The subsequent sections are organized in the following ways. Related work is discussed in section 2, and presentation of our proposed approach is detail in section 3. Section 4 shows the experimental result and related illustrations. Finally, the conclusion is discussed in the last section.

2. Related Work

In this section, we discuss about the recent cutting-edge approaches related with visual object tracking. A visual object tracking technique functionally entails with two core components: a motion model that projects the set of probable object position in the current frame by learning the estimation from the previous frames (e.g., Kalman filter [19] and particle filter [33, 34]); and an observation model that verifies the probable candidate regions based on the appearance information of the target for fixing the target position [35].

In the context of the observation model, tracking algorithms are broadly characterized as a generative or discriminative mode. In the generative approach, the tracking task is formulated as searching for the image regions most similar to the target model. A decent number of tracking algorithms have been proposed that follow the generative approach including template-based [19, 36, 37], sparse representation [38, 39], density approximation [40, 41], and cumulative subspace learning [42]. In the discriminative approach, the target object and background distinguishable model were built. These types of tracking algorithms characteristically learn classifiers based on multiple instance learning [43], P-N learning [23], online boosting [44, 45], and so on.

In the last couple of years, correlation filters have increased focus in visual tracking areas in terms of computational efficiency and enviable performance. Initially, CF(correlation filters) were inept for online tracking due to training limitations but the problem has largely been solved after the development of MOSSE [46] filter that was capable of adaptive training, and [47], a fast correlation filter capable of running on 100 fps that was designed with MOSSE filter. Another online adaptive tracker STRUCK [17] uses a kernelized structured output support vector machine to avoid the intermediate classification step and prevents to many training data through online learning and budgeting mechanism. Henriques et al. [15] articulated kernelized correlation filters (KCF) via circular matrices, and multi-channel features which are incorporated in a Fourier domain efficiently. A notable number of trackers have been developed considering KCF as baseline tracker including [48, 49]. Danelljan et al. [48] proposed an approach DSST(Discriminative Scale Space Tracking) that can estimate scaling and translation using separate independent correlation filters. The temporal memory model based tracker MUSTer [49] performances are satisfactory in constrained environments, but their low-level hand-designed features selection is susceptible

in a dynamic environment including lighting variations, occlusion, deformations, etc. In [50] the authors proposed a discriminative object appearance model built on color representation and [20] presents a scale adaptive mean-shift tracking algorithm, both lightweight methods demonstrate noteworthy performance with vigilantly selected color features that are also suitable for deformable objects. A single-object tracker STAPLE(Sum of Template and Pixel-wise Learners) is introduced in [51], which used correlation filters to handle illumination changes and a color model to handle shape deformations. The P-N learning based framework TLD [23] aims to tackle the drift problem that arises during tracking by evaluating the detector in every frame and continually updating the model till the last frame.

Despite this, CNN based algorithms are successful in many promising tasks of computer vision. For instance, image classification, object detection, object recognition and many more, as yet only a few tracking algorithms use these CNNs representations [52, 53, 54]. Early CNN based tracking algorithm [55] was limited to track only predefined target classes, [53] suffered performance compare to tracker based on hand-crafted features due to lack of training data. Approaches in [52, 54] are trained on large dataset, but their performance is satisfactory only for classification and not tracking. A CNN based multi-domain learning network [24] trained on data originated from different domain and perform domain-independent online visual tracking. In [25] a CNN tracker C-COT is proposed where the learning problem is presented in the continuous spatial domain through an implicit interpolation model. This method gains superior results in object tracking but very slow at the test time (1 fps on GPU). Martin et al. [56] uses the perception of C-COT [25] to reduce the algorithmic redundancy by combining the deep features with the hand-crafted features. Generally, Siamese network-based methods perform learning by exploiting the variations of object appearances and try to yield the similarities between target templates and candidate templates. GOTURN [26], SINT(Siamese Instance Search) [57], YCNN [58] are some of the notable existing Siamese-based trackers. A generic object tracker GOTURN [26] deals with the limitations of the C-COT [25] algorithm. GOTURN can track generic objects in real-time (100 fps on GPU) by learning the object's motion and appearance relationship in an offline manner. Even though this tracker can handle complex challenges in data like rotations, illumination changes, and viewpoint changes, it still does not perform well in cases of long-term occlusions and large movements of the target.

In this paper, we devise an object tracking algorithm fusing a color histogram-based object tracker mean-shift with GOTURN to improve the tracking accuracy. We use the ground-truth bounding box information of target objects in the first frame and the target is tracked by looking for the best-matched region using offline trained appearance and/or motion model and color histogram of our searching area. The proposed system can handle the occlusion and large movement problem with greater accuracy than GOTURN [26].

Over the years, researchers proposed a lot of approaches for image depth estimation assuming the availability of multiple observations of the scene of interest. To overcome these shortcomings [59, 60], the monocular depth estimation problem is considered as a supervised learning problem where each image pixel depth is directly predicted using an offline trained model. So, these methods are not feasible for vast applications because a huge ground-truth depth data is required. Godard et al. [27] approach depth estimation in an unsupervised way where training is accomplished like an image reconstruction problem and their fully convolutional model trained to synthesize depth as an intermediate rather than any depth data. Because of our system solely depends on a single image without any information about the geometry of scene and object types that is why we focus primarily on monocular depth estimation.

3. Proposed Approach

This section describes the details of our proposed architecture regarding the tracking algorithm for visual object tracking and depth(z) estimation for mapping audio to 3D objects. We employ the GOTURN algorithm which is a CNN model-based object tracker and the mean-shift tracking algorithm is fused with GOTURN to give the color information and lower the probability of object tracking failure. We train the classifier based on the dataset, VOT2014, and VOT2015 used in the VOT (Visual Object Tracking) Challenge [28, 29]. We then used this classifier to choose the tracker by comparing the histogram similarity of the objects appearing in the current frame during the test time. Our proposed approach is portrayed in **Fig. 3**.

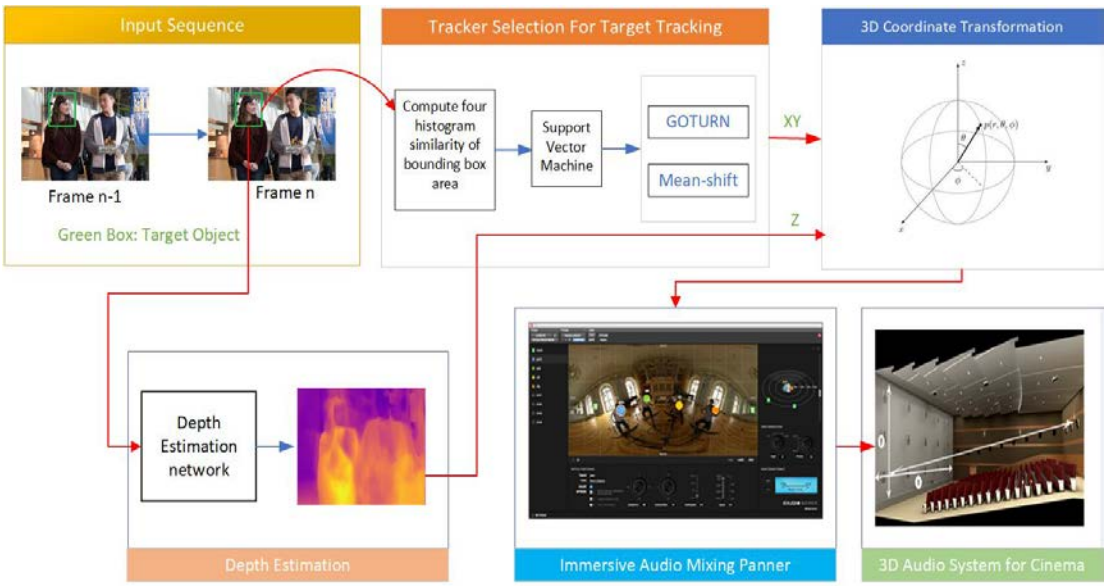


Fig. 3. Proposed approach of visual object tracking and depth estimation for automatic mixing of immersive audio into cinematography

Our working process comprises the following main steps:

1. Marking the intended object in the current frame through bounding-box as an input to the system.
2. Calculating the four-color histogram similarity of the bounding-box area marking in the current frame by running two trackers used in this system.
3. The histogram values are pass to the already trained SVM classifier to choose a tracker between two trackers GOTURN and mean-shift for tracking. The tracker will return the x and y positions of the corresponding object.
4. Estimate the depth(z) which is basically the distance between the observer position and object position in the image plane via outperforming unsupervised monocular depth estimation algorithm.
5. After getting the object 3D trajectories (x, y, z), transform to spherical (r, θ, φ) coordinates using the following formula:

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \tan^{-1}\frac{y}{x}$$

$$\varphi = \tan^{-1}\frac{\sqrt{x^2+y^2}}{z}$$

6. Using one of the object-based audio formats for mixing the specified soundtrack to the tracked object.

Our proposed approach mainly focused on visual object tracking and depth estimation for extracting the 3D positions of an object and finally, 3D audio mapping to the corresponding object, which are elucidated in detail in the following subsections.

3.1 Classification

In our paper, we alternate between two trackers to improve the accuracy of the tracking. By comparing the similarity between the histogram of the current frame bounding box with the histogram of the base frame and the third previous frame, the system chooses the tracker. We deploy SVM (Support Vector Machine) as a binary classifier to choose one tracker between the GOTURN [26] and mean-shift based tracker [20]. SVM is an algorithm for finding optimal linear boundaries that linearly separate the data to be classified based on the labeled training data. The SVM outputs an optimal hyperplane which classifies new examples. For the binary classification the optimal hyperplane is, $w^T x + b = 0$ and every data points on the 2D space satisfy the following classification criterion:

$$y_i(w^T x_i + b) \geq 1 \tag{1}$$

Where $y_i \in \{1, -1\}$ is the label of feature vector $x_i \in R^n$ and $(x_i, y_i), i = 1 \ldots . m$ refers to the training data set.

The main strength of SVM is calculating the hyperplane for the higher dimensional space using the kernels for the non-linear separable data-set [61]. Some of the kernel are as follows:

$$\text{Polynomial kernel: } k(x, y) = (\alpha x^T y + c)^d \tag{2}$$

Where the adaptable parameter α is denoted as slope, constant term is c, and the polynomial degree is d.

$$\text{Exponential kernel: } k(x, y) = exp\left(-\frac{||x-y||}{2\sigma^2}\right) \tag{3}$$

Where the parameter σ is regulating and need to be estimated carefully. An example of SVM classification is shown in **Fig. 4**, where hyperplane H_2 and H_3 classify the two categories of data sufficiently but H_3 is the optimal hyperplane which maintains the highest margin.

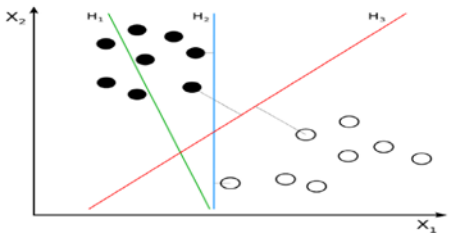


Fig. 4. An example of SVM data classification

During training, we constructed four histogram similarities and one label in the SVM learning data form. The histogram similarity is computed by the correlation method as shown in equation (4). The parameter h_1 and h_2 are two same size array of histogram and b is the total number of histogram bins.

$$S(h_1, h_2) = \frac{\sum_i (h_1(i) - \overline{h_1})(h_2(i) - \overline{h_2})}{\sqrt{\sum_i (h_1(i) - \overline{h_1})^2 \sum_i (h_2(i) - \overline{h_2})^2}}, \text{where } \overline{h_x} = \frac{1}{b}\sum_i h_x(i) \tag{4}$$

We consider four histogram similarities by comparing the histogram of the bounding box area in the first frame(t=0) and the tracked bounding box area found in the current frame(t=n) using GOTURN tracker and mean-shift tracker, and comparing the histogram of the third previous frame(t=n-3) and the current frame(t=n) using GOTURN and mean- shift tracker, shown in the **Fig. 5**.

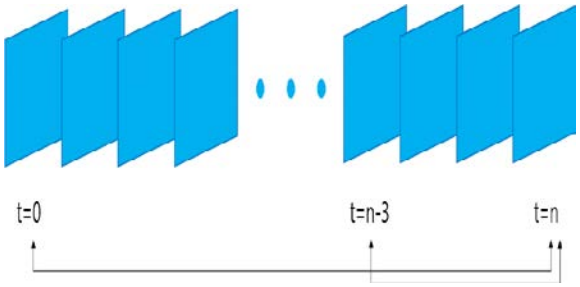


Fig. 5. Procedure of frame selection for histogram comparison

The label required for training SVM classifier is based on the value of IoU (Intersection over Union). IoU is an index to judge the accuracy of object detection by the ratio of the intersecting area and the union area of the two bounding boxes. **Fig. 6** and equation (5) depict the concept of IoU. In our experiment, the IoU values are the contrast between the resultant coordinates of the GOTURN and mean-shift tracker with the ground truth values. If both cases the IoU becomes 0, the distance of the tracker and ground truth is chosen as the label.

$$IoU = \frac{A_G \cap A_T}{A_G \cup A_T} \tag{5}$$

Where A_G denotes the area of the ground-truth bounding box and A_T denotes the area of the tracked bounding box.

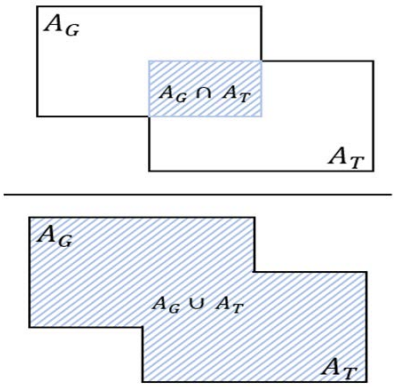


Fig. 6. Intersection over union (IoU)

Fig. 7 shows the tracker selection procedure. We use SVM to learn data using histogram similarity and labels using IoU and choose whether to use GOTURN or mean-shift tracker for the tracking. After selecting the tracker by the SVM we calculate the histogram similarity of the tracked area and compare it with a threshold value to update the value of the trackers for future tracking. Besides, the mean-shift tracker traces the object using only the initial histogram model, which causes the bounding box to jump to a place other than the tracked object. In **Fig. 7**, the movement distance of the bounding box is named mean-shift distance. If the bounding box moves beyond the threshold value in the previous frame and the current frame by giving a threshold value according to the resolution of the image, the GOTURN is initialized at the previous bounding box position. The higher the resolution of the input image, the greater the change in the position of the bounding box when the object moves. Therefore, the threshold proportional to the input image size is set as shown is equation (6), where I_w and I_h are width and height of the input image I, respectively.

$$Threshold = \sqrt{{I_w}^2 + {I_h}^2}/100 \tag{6}$$

And if the brightness level of the image is greater than 8 or less than 3, it is initialized with GOTURN to increase the tracking accuracy. The brightness is linearly divided into 1 ~ 10 levels.

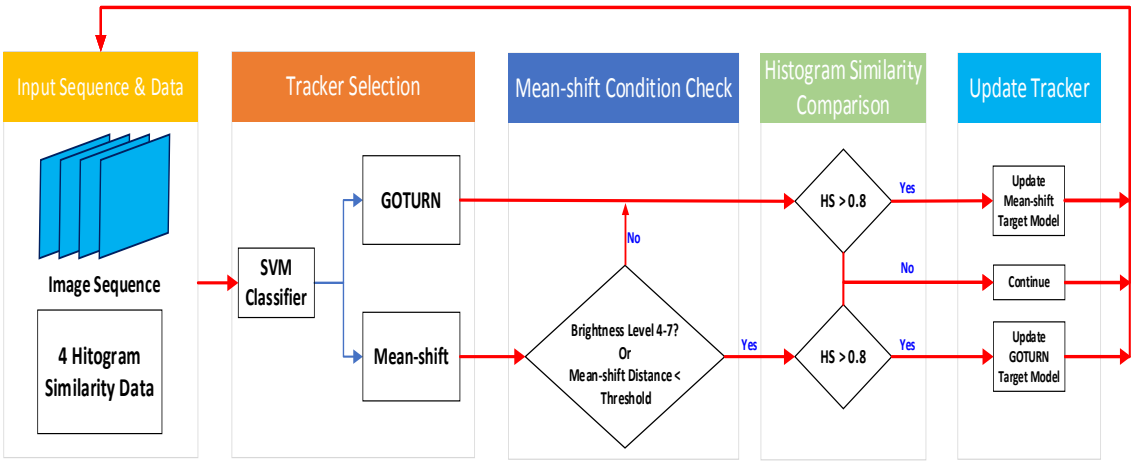


Fig. 7. The tracker selection and model updating procedures

3.2 Tracker

Trackers are used to locate novel objects marked in one frame of a video into subsequent video frames and maintain their trajectories till the end. In this work, we design a tracking algorithm via two existing prominent single object tracker GOTURN and mean-shift that exhibits better performance in visual object tracking. The subsequent parts cover the concept of these tracking algorithms.

3.2.1 GOTURN

The leading real-time tracking capable CNN based generic object tracker is GOTURN, a model trained from several labeled videos and images not having the use of the target objects class labels or types being tracked. The GOTURN framework generically develops the object's appearance and motion relationship to the network training in an offline manner which helps the network to run in real-time [26]. The network architecture of GOTURN is described in **Fig. 8**.

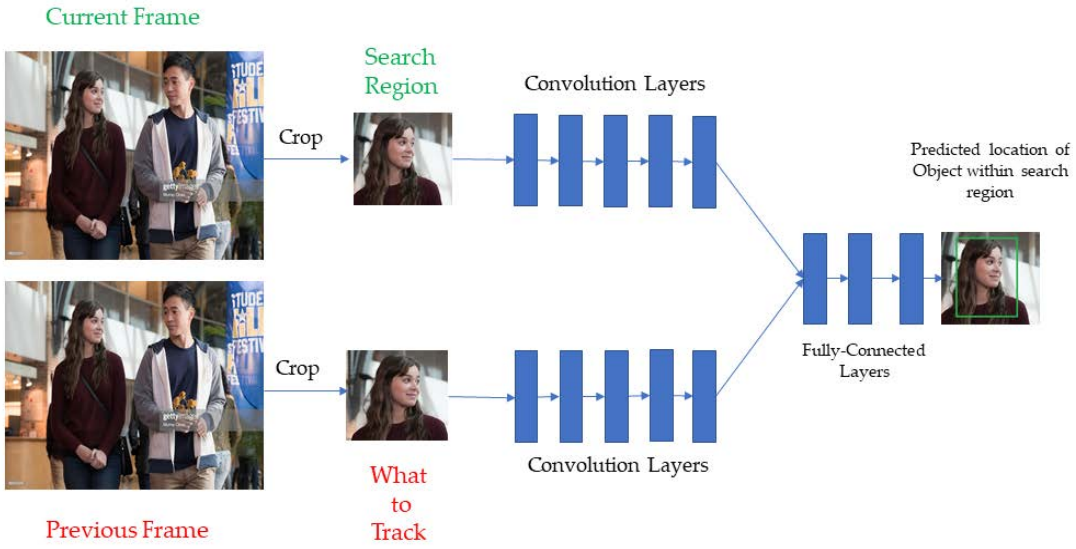


Fig. 8. GOTURN network architecture for single object tracking.

During operation, the human observer initializes the tracker with the first frame bounding box information. At each subsequent frame *n*, the network receives the image crops from frame *n*-1 and frame *n* respectively as an input to predict the object location in the n^{th} frame. we select randomly contiguous frame pairs in the learning phase, cut the object region to be tracked in the previous frame, cut the same area in the current frame, and learn information about the object and its surroundings in the convolution layer. Thus, we learn how to predict the bounding box position in the current frame by sharing two convolution layer weight values in the connected layer.

Therefore, it is possible to track an object at high speed using only offline learning data. However, if there is no motion information and some of the tracking objects are obscured by other objects, the performance is significantly degraded. Also, once a trace fails, it will continue to track based on the location of the failed trace. Therefore, in this study, we decided that the initial set object will be tracked again if the GOTURN has the color information of the initial setting object even when the tracking fails. To compensate for the disadvantages mentioned above, the mean-shift algorithm is used.

3.2.2 Mean-shift

Mean-shift is a method of locating the mean of data distribution, in which data is moved in the densest direction around the current one. When the mean-shift is used in object tracking, the color histogram of the specified object is compared with the input image histogram to find the region having the most similar histogram. In the mean-shift based real-time tracking approach [62], the Bhattacharyya coefficient metric is used for target localization. The maximum Bhattacharyya coefficient value gives highest similarity among the pdf (probability density function). The probability density function of the target model and target candidate is represented by equation (7) and (8). An example of mean-shift tracking is visualized in **Fig. 9**.

$$\text{Target model:} \quad \vec{q} = \{\vec{q}_u\}_{u=1\ldots m}, \ \sum_{u=1}^{m} \vec{q}_u = 1 \tag{7}$$

$$\text{Target candidate:} \quad \vec{p}(y) = \{\vec{p}_u(y)\}_{u=1\ldots m}, \sum_{u=1}^{m} \vec{p}_u = 1 \tag{8}$$

In these equations, m is the number of discrete bin used for color distribution in histogram, $\vec{q}_u$ is the density function of the color/texture feature u of target model centered at origin, and $\vec{p}_u(y)$ represents the candidate density centered at location y.

The Bhattacharyya coefficient is expressed in equation (9), which is a similarity measure between the distribution target model $\vec{q}$ and target candidate $\vec{p}(y)$ at location y.

$$\text{Bhattacharyya coefficient:} \ \rho[\vec{p}(y), \vec{q}] = \sum_{u=1}^{m} \sqrt{\vec{p}_u(y)\vec{q}_u} \tag{9}$$

The Hellinger distance estimates the difference between two probability distributions $\vec{q}$ and $\vec{p}(y)$.

$$\text{Hellinger distance:} \ H(\vec{p}(y), \vec{q}) = \sqrt{1 - \rho[\vec{p}(y), \vec{q}]} \tag{10}$$

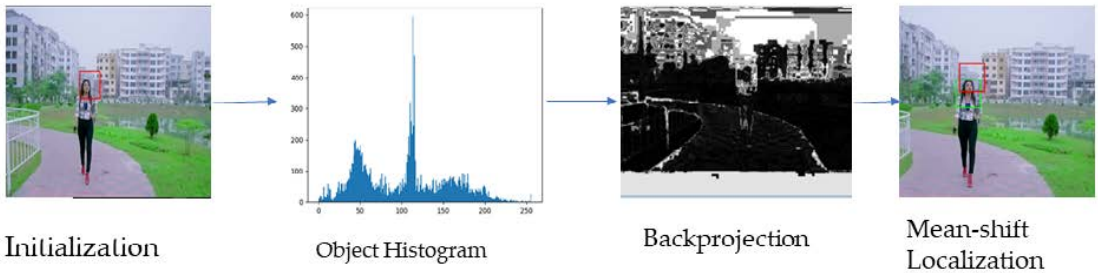


Fig. 9. Visual procedure of mean-shift tracker

In this paper, we apply a robust scale-adaptive mean-shift tracking algorithm [20], which is flexible in color-based object tracking and search radius. This algorithm is the same as finding the center point of the mean-shift tracker. However, after searching the center point, it changes the scale of the window size and finds the window size with the highest histogram similarity degree of the specified object. It shows higher performance than the conventional mean-shift tracker.

3.3 Depth Estimation

To mix a soundtrack to a particular object, we need to estimate the third dimension, the depth(*z*) of that object. We are looking for a depth estimation method where only one image is taken as input without any ground-truth depth data. Godard et al. [27], proposed an end-to-end monocular depth estimation network where in the convolution neural network is trained with an innovative loss function that enacts left-right consistency among the disparity images not having the help of ground truth data supervision. Its operation is described in the following steps:

1) *Depth calculation in the form of image reconstruction*

In the time of training, the perception of depth calculation by way of image reconstruction is that in the presence of two rectified binocular cameras by learning a function an image can reconstruct from another image that will aid to learn the 3D geometry of the scene. Then using this reconstruction information, the system can forecast the image pixel depth, $\hat{d} = {}^{bf}/_{d}$, where, d is a scalar quantity belongs to every pixel denoted as image disparity, and b is the physical space among the cameras and camera focal length f.

2) *Depth Estimation Network*

A bilinear sampler is used in this network to generate the predicted image with backward mapping resulting in a completely differentiable image construction model as depicted in **Fig. 10**.

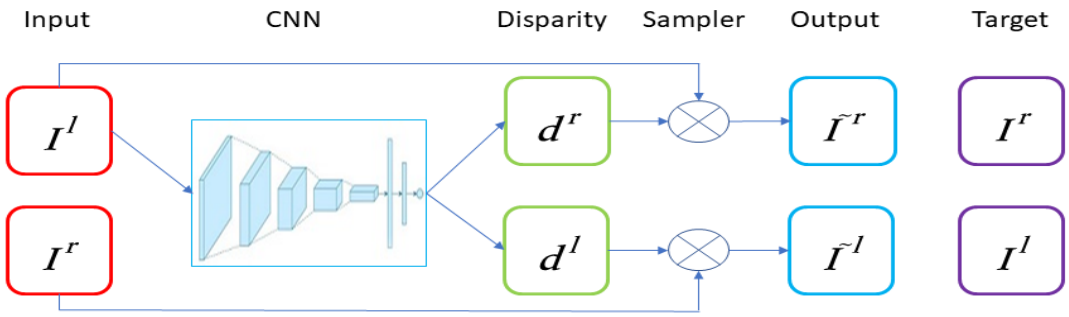


Fig. 10. Backward mapping using left images to make disparities for left and right image

3) *Training Loss*

The total training loss C by combining a loss C_s at different scale s, establishing the equation, $C = \sum_{s=1}^{4} C_s$ and C_s as a combination of three main components, appearance matching loss, disparity smoothness loss, and left-right disparity consistency loss.

$$C_s = \alpha_{ap}\left(C_{ap}^l + C_{ap}^r\right) + \alpha_{ds}\left(C_{ds}^l + C_{ds}^r\right) + \alpha_{lr}\left(C_{lr}^l + C_{lr}^r\right) \quad (11)$$

Where C_{ap} inspires the similar appearance of the input and reconstructed images, C_{ds} imposes disparity smoothing, and C_{lr} denotes the consistency between left-right image disparities.

The visual output of the dense depth estimation of a particular scene using Godard et.al [27] algorithm is shown in **Fig. 11**.

Fig. 11. Qualitative results on a particular video scene with thin structures

4) *Extract object depth value*

The x and y coordinates can be obtained by taking the center coordinates of the bounding box, but the depth value is quite erroneous when obtaining the values of the center coordinates due to the noise that is often generated. Therefore, in this paper the average value of all depth values inside the bounding box estimated by the tracker is extracted as the depth value(D).

$$D = \sum_{i \in Bbox} d_i \; / \sum_{i \in Bbox} i \tag{12}$$

Where d_i denotes depth value of the pixel i in the bounding box.

4. Experimental Results

In this section, we will show the visual object tracking results applying our proposed tracker and the qualitative result of depth estimation to make immersive audio founded cinema and rigorously analyze the results. A deep learning framework Caffe is used for the experiment. The experimental computer specifications are as follows: CPU is Intel corei7-7700@ 3.60GHz x 8, GPU is Nvidia GeForce GTX 1080 Ti/PCIe with cuDNN acceleration, and the memory is 16 GB. The underlying operating system is Ubuntu 16.04 LTS and RGB-D sensors for assisting depth estimation. In our experiment, we use VOT2014 [28] and VOT2015 [29] dataset from visual object tracking challenges. The dataset comprises 25 and 60 sequences with objects in challenging backgrounds and selected from the popular ALOV, OTB2, and some non-tracking datasets. To evaluate the object tracking performance of the proposed algorithm, we employ IoU as an evaluation metric. The details are depicted in section 3.1.

4.1 Results

As a test data set, we used VOT2014 benchmark with 25 video sequences used in the Visual Object Tracking VOT2014 Challenge [28]. We compared our proposed technique with four other state-of-art trackers: GOTURN [26], TLD [23], Mean-Shift [20] and STRUCK [17]. The total number of video frames is 10214 in the VOT2014 benchmark. We consider tracking of an object successful when the value of IoU is greater than 50% otherwise it is considered as unsuccessful tracking.

For the performance analysis the average IoU was calculated on each sequence of the VOT2014 benchmark by the aforementioned trackers: GOTURN, TLD, Mean-Shift, and STRUCK. **Table 1** and **Fig. 16** present the tracking performance achieved on each of the 25

video sequences of the VOT2014 challenging dataset through the trackers declared here along with the proposed tracker. The number of frames successfully tracked by each tracker is as follows: GOTURN (6151), TLD (1578), Mean-shift (2960), STRUCK (2950) and Ours (6178). Therefore, the overall accuracy of each trackers is as follows: GOTURN (48.88%), TLD (19.39%), Mean-shift (30.91%), STRUCK (28.79%) and Ours (50.50%) respectively. From the experimental results it is evident that our proposed tracker exhibits the best performance among the above-mentioned trackers.

We also tested our proposed algorithm on a cinema video and 360-degree drama to prove the rationality of this work. The testing results are shown in **Fig. 12** and **Fig. 13**. We can see that the 3D coordinates of the object selected at the first frame are continuously and accurately extracted. Additionally, some example video sequences are shown in **Fig. 14** and **Fig. 15**, where the performance is significantly improved and degraded over the other four trackers.

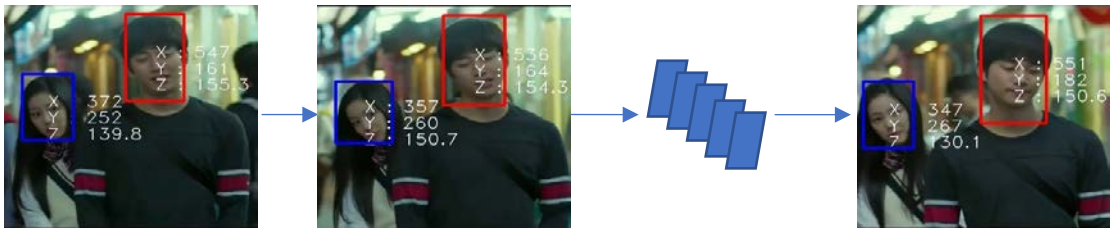


Fig. 12. The extracted 3D positions of the target object in the successive frames

Fig. 13. The extracted 3D positions of the target object in the 360-degree drama frames

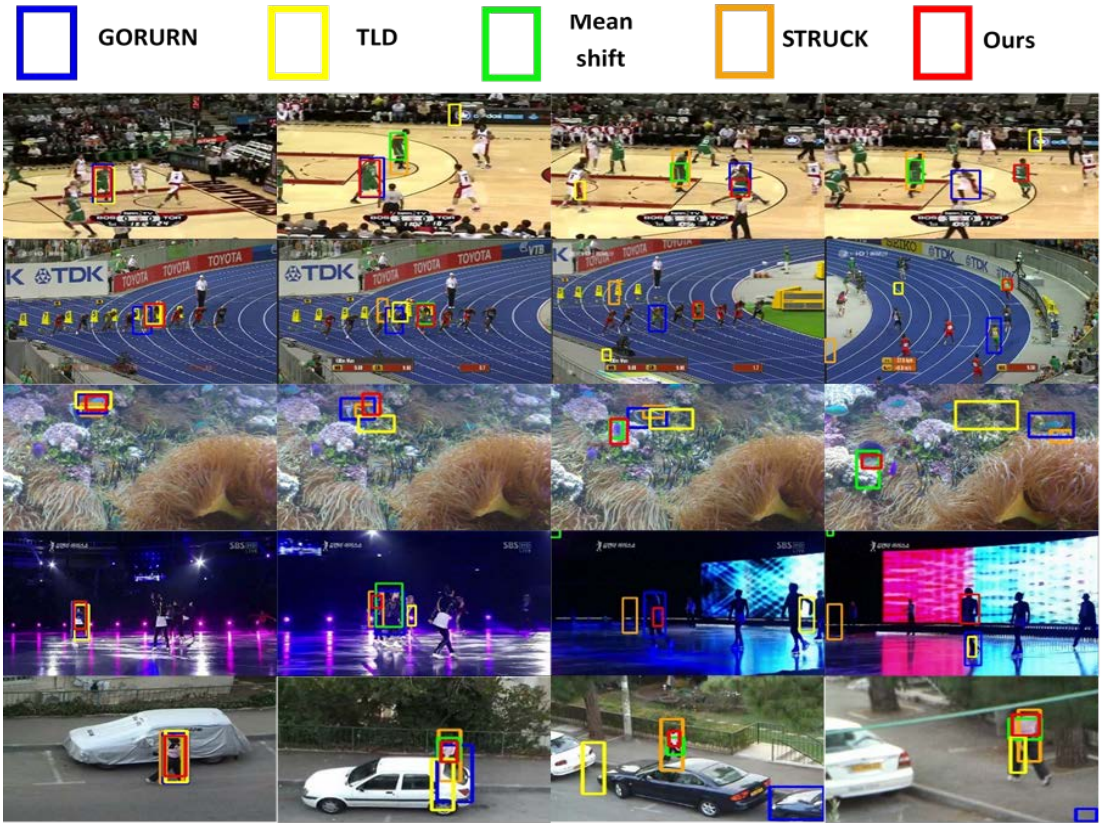


Fig. 14. Examples of improved performance compared to other four trackers on VOT2014 benchmark

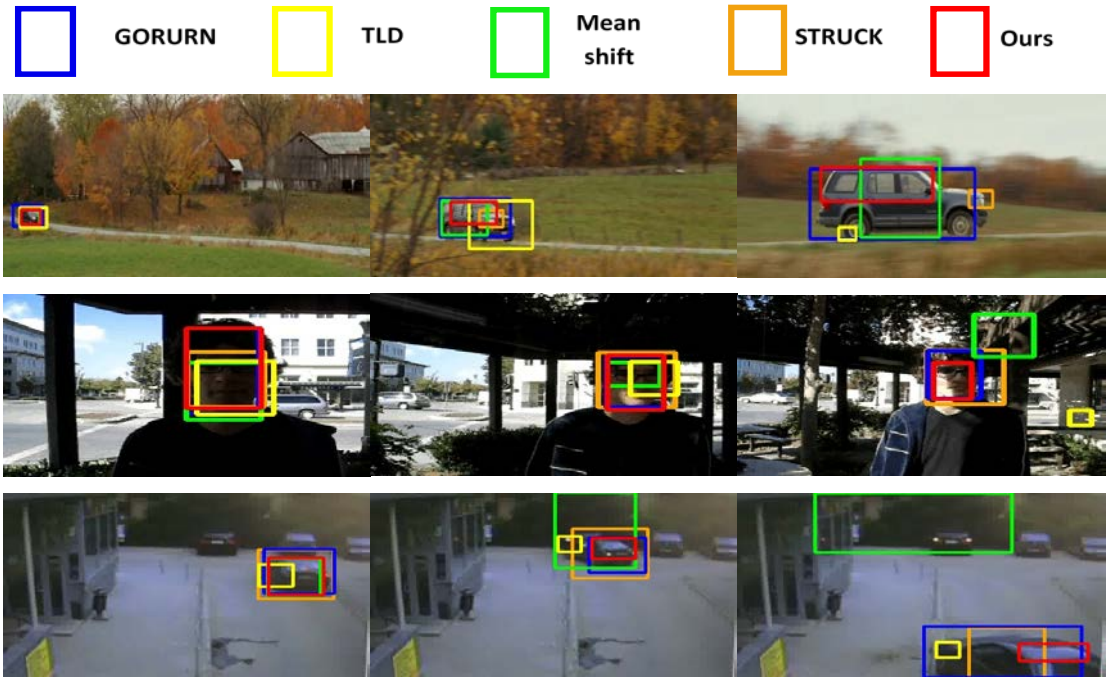
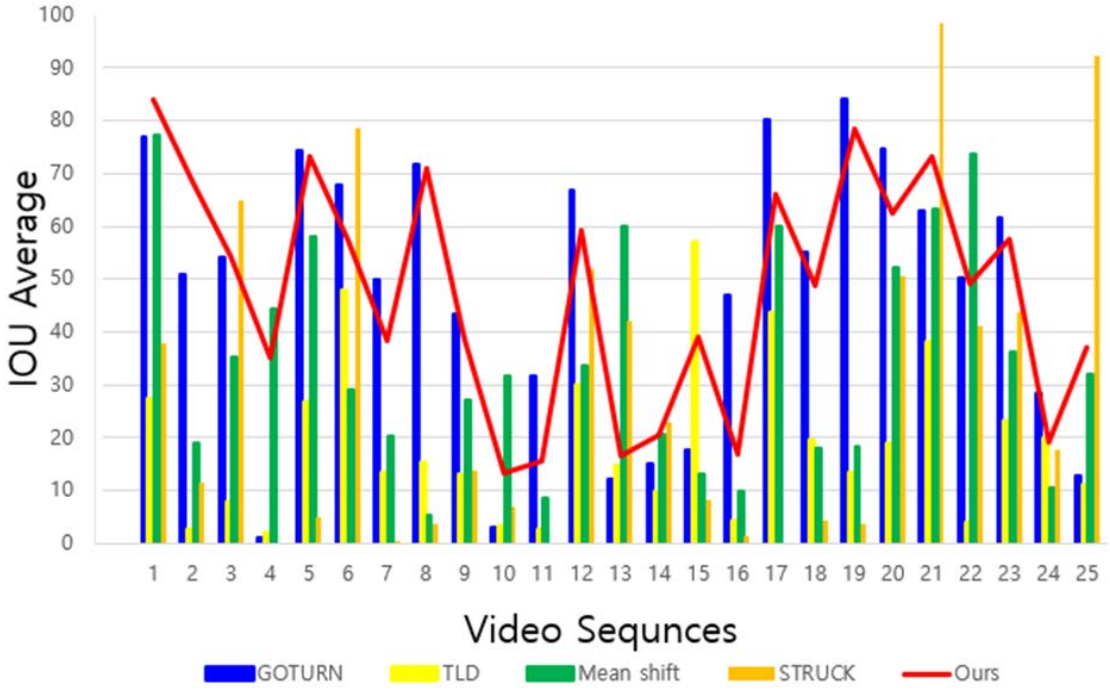
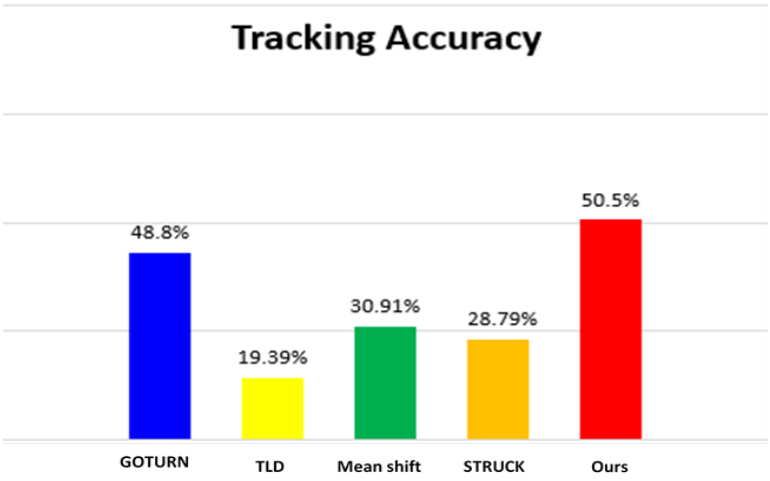


Fig. 15. Examples of degraded performance compared to some other trackers from VOT2014 benchmark



(a)



(b)

Fig. 16. **(a)** Video sequence wise IoU average comparison of VOT2014 challenging dataset **(b)** The overall tracking accuracy of compared trackers and our proposed tracker on VOT2014 challenging dataset

Table 1. Performance comparison based on each video sequence

Video Sequence	Total Frame Number	IoU Average (%)					# of Successfully Tracked Frames				
		GOTURN	TLD	Mean shift	STRUCK	Ours	GOTURN	TLD	Mean shift	STRUCK	Ours
1	603	76.93	27.36	77.26	37.61	83.78	596	98	600	214	600
2	725	50.7	2.43	18.72	11.18	68.23	474	6	71	98	631
3	271	54.13	7.68	35.26	64.79	54.4	159	0	131	224	164
4	350	0.78	1.93	44.33	0	34.97	2	11	140	2	141
5	252	74.06	26.71	58.01	4.96	73.09	252	36	178	2	242
6	770	67.74	47.64	28.95	78.42	57.42	749	366	64	696	627
7	219	49.83	13.33	20.17	0.08	38.43	81	32	22	2	43
8	1210	71.48	15.27	5	3.48	70.77	1105	34	28	2	1079
9	292	43.16	13.02	26.8	13.61	38.77	150	10	40	23	75
10	436	2.8	3.31	31.54	6.72	13.42	5	4	93	28	29
11	310	31.44	2.47	8.39	0	15.52	72	5	5	2	51
12	207	66.8	29.96	33.48	51.83	59.27	184	69	59	88	148
13	244	12.09	14.53	59.98	41.94	16.44	16	29	174	113	42
14	267	14.76	9.55	20.58	22.89	20.46	28	12	35	54	52
15	307	17.64	56.9	12.99	8	39.09	69	209	62	7	169
16	164	46.82	4.12	9.74	1.23	16.77	98	7	12	4	27
17	371	79.89	43.54	59.83	0	65.92	371	177	318	2	348
18	400	54.83	19.58	17.95	4.26	48.73	239	26	21	6	226
19	201	83.91	13.39	18.24	3.65	78.45	201	28	23	8	167
20	172	74.53	18.85	51.92	50.53	62.58	172	38	85	81	163
21	282	62.72	38.09	63.05	98.31	73.18	243	50	269	282	257
22	264	50.17	3.72	73.39	40.9	48.97	172	11	218	124	161
23	569	61.62	23.01	36.18	43.57	57.72	517	142	209	347	406
24	731	28.29	19.93	10.3	17.5	19.12	86	135	81	9	26
25	597	12.76	10.89	31.91	92.11	37.08	110	43	22	549	304

5. Conclusion

In this work, we devise an object tracking algorithm fusing CNN and color histogram-based trackers that extracts the *XY* coordinates of the target object. Then, we estimate the depth(*Z*) position of the target via an unsupervised monocular depth estimation algorithm heading to make automatic immersive audio founded cinema. To upsurge the tracking accuracy and handle the occlusion problems of the GOTURN algorithm, we consolidate CNN based GOTURN algorithm and a color histogram-based mean-shift tracking algorithm. The SVM classifier is then used to select a tracker with higher tracking accuracy. The performance of our tracker is improved by selecting one of the two trackers based on the histogram similarity. The overall tracking accuracy of the proposed algorithm is about 2% and 19% improved than the existing cutting-edge GOTURN and mean-shift algorithm. In the

current object tracking research, as with our proposed algorithm, occlusion and re-identification areas do not perform robustly in the sequences of various domains. Thus, many researchers are still actively researching to solve these tracking issues. Also, since the bounding box does not completely represent the tracking object, there is a problem that noise due to the background is included when calculating the histogram similarity. In future, we will research on object segmentation to improve the accuracy of the tracking algorithm.

Acknowledgements

This research was supported by the GRRC program of Gyeonggi province [GRRC Aviation 2018-B04, Development of Interactive VR Player and Service with Space Media Convergence].

References

[1] S. Sivanantham, N. N. Paul, and R. S. Iyer, "Object tracking algorithm implementation for security applications," *Far East Journal of Electronics and Communications*, vol. 16, no. 1, pp. 1-13, 2016. Article (CrossRefLink)

[2] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Yongdong Zhang, and Q. Dai, "STAT: Spatial-Temporal Attention Mechanism for video Captioning," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 229-241, 2020. Article (CrossRefLink)

[3] J. Severson, "Human-digital media interaction tracking," US Patent 9,713,444, 2017. Article (CrossRefLink)

[4] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *Proc. of Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. IEEE*, pp. 1103–1108, 2011. Article (CrossRefLink)

[5] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, "Safe driving envelopes for path tracking in autonomous vehicles," *Control Engineering Practice*, vol. 61, pp. 307-316, 2017. Article (CrossRefLink)

[6] V. A. Laurense, J. Y. Goh, and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," in *Proc. of American Control Conference (ACC), IEEE*, 2017. Article (CrossRefLink)

[7] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, 38(4), 2006. Article (CrossRefLink)

[8] T. Zhang, K. Jia, C. Xu, Y. Ma, and N. Ahuja, "Partial occlusion handling for visual tracking via robust part matching," in *Proc. of* the *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1258–1265, 2014. Article (CrossRefLink)

[9] J. Pan and B. Hu, "Robust occlusion handling in object tracking," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007. Article (CrossRefLink)

[10] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 11, pp. 1531–1536, 2004. Article (CrossRefLink)

[11] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity based collaborative model," in *Proc. of IEEE Conference on Computer vision and pattern recognition(CVPR)*, pp. 1838–1845, 2012. Article (CrossRefLink)

[12] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. of IEEE Conference on Computer vision and pattern recognition(CVPR)*, pp. 798–805, 2006. Article (CrossRefLink)

[13] B. Babenko, M.H. Yang and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983-990, 2009. Article (CrossRefLink)

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," *Proceedings of the IEEE*, 86(11), pp. 2278-2324, 1998. Article (CrossRefLink)
[15] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Highspeed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell*., 37(3), pp.583–596, 2015. Article (CrossRefLink)
[16] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense Spatio-temporal context learning," in *Proc. of European Conference on Computer Vision*, pp. 127-141, 2014. Article (CrossRefLink)
[17] S. Hare, A. Saffari and P.H. Torr, "Struck: Structured output tracking with kernels," in *Proc. of IEEE International Conference on Computer Vision*, pp. 263-270, 2011. Article (CrossRefLink)
[18] Doucet, D. N. Freitas, and N. Gordon, *Sequential Monte Carlo Methods*, Practice, Springer, New York, 2001. Article (CrossRefLink)
[19] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), pp. 564-577, 2003. Article (CrossRefLink)
[20] T. Vojir, N. Jana and M. Jiri., "Robust scale-adaptive mean-shift for tracking," *Pattern Recognition Letters*, vol. 49, pp. 250-258, 2014. Article (CrossRefLink)
[21] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. of NIPS*, pp. 809-817, 2013. Article (CrossRefLink)
[22]N. Wang, J. Shi, D.Y. Yeung and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proc. of 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. Article (CrossRefLink)
[23] Z. Kalal, K. Mikolajczyk and, J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34(7), pp.1409-1422, 2012. Article (CrossRefLink)
[24] Nam, H. Hyeonseob and Bohyung, "Learning Multi-Domain Convolutional Neural Networks for Visual Tracking," in *Proc. of The IEE Conference on Computer Vision and Pattern Recognition*, June 2016. Article (CrossRefLink)
[25] M. Danelljan, A. Robinson, F.S. Khan and M. Felsberg, "Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking," in *Proc. of the European Conference on Computer Vision(ECCV)*, pp. 472-488, 2016. Article (CrossRefLink)
[26] D. Held, S. Thrun, and S. Savarese, "learning to track at 100 fps with deep regression networks," in *Proc. of European Conference on Computer Vision, Springer, Cham*, pp. 749-765, October 2016. Article (CrossRefLink)
[27] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017. Article (CrossRefLink)
[28] M. Kristan, R. Pugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojir, G. Fernandez, A. Luke_zi_c, A. Dimitriev, et al., "The visual object tracking VOT2014 challenge results," in *Proc. of Computer Vision-ECCV 2014 Workshops, Springer*, pp. 191-217, 2014. Article (CrossRefLink)
[29] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, and R. Pflugfelder, "The visual object tracking VOT2015 challenge results," in *Proc. of the IEEE international conference on computer vision workshops*, pp. 1-23, 2015. Article (CrossRefLink)
[30] L. Leal-TaixÂ´e, C. Canton-Ferrer, and K. Schindler, "Learning by tracking: Siamese CNN for robust target association," in *Proc. of CVPRW*, 2016. Article (CrossRefLink)
[31] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid and K. Schindler, "Online multi-target tracking using recurrent neural networks," *Computer Vision and Pattern Recognition (cs.CV)*, 2016. Article (CrossRefLink)
[32] B. Wang, L. Wang, B. Shuai, Z. Zuo, T. Liu, K. Luk Chan and G. Wang, "Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association," in *Proc. of CVPRW*, 2016. Article (CrossRefLink)
[33] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, "Color-based probabilistic tracking," in *Proc. of* the *European Conference on Computer Vision*, pp. 661–675, 2002. Article (CrossRefLink)

[34] Y. Li, H. Ai, T. Yamashita, S. Lao, M. Kawade, "Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30(10), pp.1728–1740, 2008. Article (CrossRefLink)
[35] X. Li, W. Hu, C. Shen, Z. Zhang, A.R. Dick, A. van den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4(4), pp.1–48, 2013. Article (CrossRefLink)
[36] J. Kwon, K.M. Lee, "Tracking by sampling and integrating multiple trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36(7), pp.1428–1441, 2014. Article (CrossRefLink)
[37] N. Wang, J. Wang, and D.-Y. Yeung, "Online Robust Non-negative Dictionary Learning for Visual Tracking," in *Proc. of ICCV, 2013.* Article (CrossRefLink)
[38] X. Mei and H.Ling, "Robust visual tracking using l_1 minimization," in *Proc. of ICCV*, 2009. Article (CrossRefLink)
[39] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. of CVPR*, 2012. Article (CrossRefLink)
[40] B. Han, D. Comaniciu, Y. Zhu, and L. Davis, "Sequential kernel density approximation and its application to real-time visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30(7), pp.1186-1197, 2008. Article (CrossRefLink)
[41] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25(10), pp.1296–1311, 2003. Article (CrossRefLink)
[42]D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *IJCV*, 77(1-3), pp.125–141, 2008. Article (CrossRefLink)
[43] B. Babenko, M. H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33(8), pp.1619–1632, 2011. Article (CrossRefLink)
[44] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. of BMVC*, pp. 6.1-6.10, 2006. Article (CrossRefLink)
[45] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. of ECCV*, pp. 234-247, 2008. Article (CrossRefLink)
[46] D. S Bolme, J R. Beveridge, B. A Draper, and Y. M Lui. 2010, "Visual object tracking using adaptive correlation filters," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2010. Article (CrossRefLink)
[47] Z. Cai, L. Wen, J. Yang, Z. Lei, and S. Z. Li, "Structured visual tracking with dynamic graph," in *Proc. of ACCV*, pp. 86-97, 2012. Article (CrossRefLink)
[48] M. Danelljan, G. H¨ager, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. of BMVC*, 2014. Article (CrossRefLink)
[49] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-Store Tracker (MUSTer): a cognitive psychology inspired approach to object tracking," in *Proc. of CVPR*, 2015. Article (CrossRefLink)
[50] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model free tracking," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2015. Article (CrossRefLink)
[51] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P.H.S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. Article (CrossRefLink)
[52] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. of ICML*, pp. 597–606, 2015. Article (CrossRefLink)
[53] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning discriminative feature representations by convolutional neural networks for visual tracking," in *Proc. of BMVC*, 2014. Article (CrossRefLink)
[54] N.Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015. Article (CrossRefLink)

[55] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Trans. Neural Networks*, vol. 21(10), pp.1610–1623, 2010. Article (CrossRefLink)
[56] M. Danelljan, G. Bhat, F.S. Khan, M. Felsberg, "ECO: efficient convolution operators for tracking," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6638–6646, 2017. Article (CrossRefLink)
[57] R. Tao, E. Gavves, and A. Smeulders, "Siamese instance search for tracking," in *Proc. of CVPR*, pp.1420–1429, 2016. Article (CrossRefLink)
[58]K. Chen and W. Tao, "Once for all: a two-flow convolutional neural network for visual tracking," *IEEE T-CSVT*, vol. 28(12), pp. 3377-3386, 2017. Article (CrossRefLink)
[59] D. Eigen, C. Puhrsch, and R. Fergus., "Depth map prediction from a single image using a multi-scale deep network," *NIPS*, 2014. Article (CrossRefLink)
[60] F. Liu, C. Shen, G. Lin, and I. Reid., "Learning depth from single monocular images using deep convolutional neural fields," *PAMI*, vol. 38(10), pp. 2024-2039, 2015. Article (CrossRefLink)
[61] B. E. Boser, I. M. Guyon, V. N. Vapnik, "A training algorithm for optimal margin classifiers," in in *Proc. of the fifth annual workshop on Computational learning theory*, pp. 144-152, 1992. Article (CrossRefLink)
[62] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 142-149, 2000. Article (CrossRefLink)

Sung-Jun Park received his B.S degree in Avionics and Information Engineering in 2018 from the Korea Aerospace University, Goyang, South Korea. Currently he is pursuing his M.S. program under the School of Electronics and Information Engineering, Korea Aerospace University, South Korea. His research interests include image processing and computer vision.

Md. Mahbubul Islam received his Bachelor of Science (Honors) and M.S. (Engg.) degrees in Computer Science & Engineering in 2007 and 2008, respectively from the University of Chittagong, Bangladesh. Currently he is pursuing his Doctoral program under the School of Electronics and Information Engineering, Korea Aerospace University, South Korea. He presently holds the position of Assistant Professor in the Department of Computer Science & Engineering, University of Chittagong, Bangladesh. His research areas include image processing, computer vision and multimedia.

Joong-Hwan Baek received his B.S degree from the Korea Aerospace University (KAU), Goyang, South Korea in 1981, and his M.S. and Ph.D. degrees in Electrical and Computer Engineering from Oklahoma State University in 1987 and 1991, respectively. He was a senior researcher at Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea from 1991 to 1992. Since 1992, he has been a professor with the School of Electronics and Information Engineering, KAU. His current research interests include image processing, computer vision, pattern recognition, and multimedia.