

Article

Comparison of Tracking Techniques on 360-Degree Videos

Tzu-Wei Mi and Mau-Tsuen Yang * 

Department of Computer Science & Information Engineering, National Dong Hwa University,
Hualien 974, Taiwan

* Correspondence: mtyang@gms.ndhu.edu.tw

Received: 29 June 2019; Accepted: 12 August 2019; Published: 14 August 2019



Featured Application: Tracking techniques are essential for attaching an AR tag to a physical target in 360-degree videos.

Abstract: With the availability of 360-degree cameras, 360-degree videos have become popular recently. To attach a virtual tag on a physical object in 360-degree videos for augmented reality applications, automatic object tracking is required so the virtual tag can follow its corresponding physical object in 360-degree videos. Relative to ordinary videos, 360-degree videos in an equirectangular format have special characteristics such as viewpoint change, occlusion, deformation, lighting change, scale change, and camera shakiness. Tracking algorithms designed for ordinary videos may not work well on 360-degree videos. Therefore, we thoroughly evaluate the performance of eight modern trackers in terms of accuracy and speed on 360-degree videos. The pros and cons of these trackers on 360-degree videos are discussed. Possible improvements to adapt these trackers to 360-degree videos are also suggested. Finally, we provide a dataset containing nine 360-degree videos with ground truth of target positions as a benchmark for future research.

Keywords: 360-degree videos; augmented reality; real-time tracking

1. Introduction

Nowadays, 360-degree videos are becoming more and more popular. Omnidirectional cameras, also called 360-degree cameras, are widely available and more lightweight, and can even be installed on drones [1]. They are useful for recording indoor or outdoor activities to cover views in all perspectives. Rendering 360-degree videos on a virtual reality headset can provide immersive experience for users of education, entertainment, and tourism [2]. For augmented reality applications using 360-degree videos, a common request is to register a virtual tag to a physical target. As shown in Figure 1, a virtual billboard marked in red color must follow its corresponding physical target over time. For this purpose, automatic tracking of a specific target in 360-degree videos is highly desirable. Therefore, we explore the characteristics of 360-degree videos and compare the performance of existing tracking techniques on 360-degree videos.

A 360-degree video consists of a sequence of 360-degree images with a fixed interval of time. Each 360-degree image is a panorama either captured by an omnidirectional camera or combined by multiple cameras to cover the complete horizontal field of view (i.e., 360-degree FOV). As shown in Figure 1, a 360-degree image is typically flattened in an equirectangular format in that longitude lines are projected to vertical straight lines of constant spacing. Similarly, latitude lines are mapped to horizontal straight lines of constant spacing.

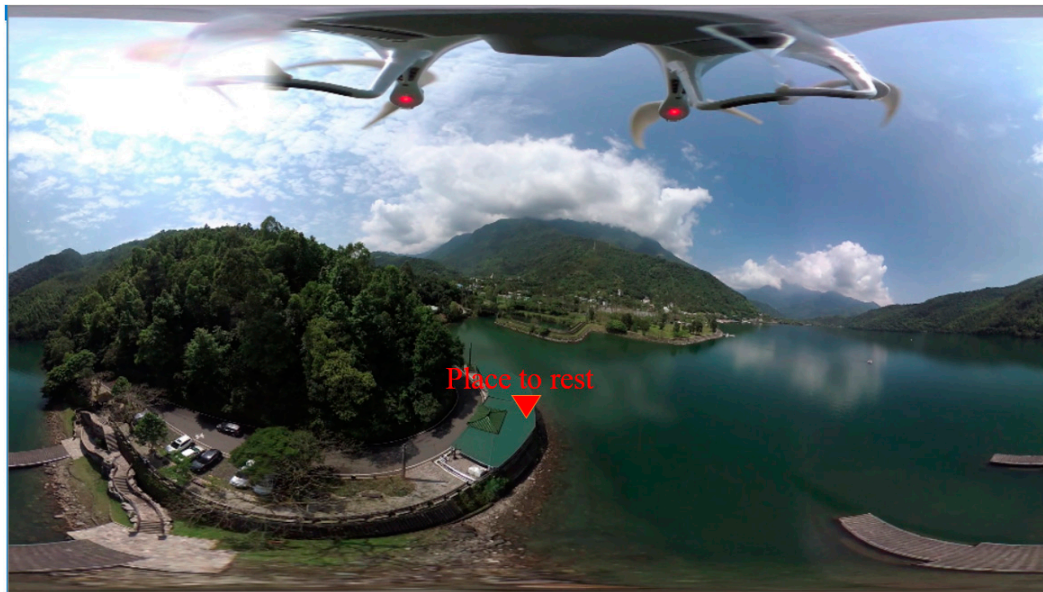


Figure 1. Tagging an Augmented Reality (AR) marker on a physical target in 360-degree videos.

Due to the natures of the immersiveness and full field of view, 360-degree videos are ideal to be adopted in virtual reality applications. However, the interaction ways during playback of 360-degree videos are quite limited. Users cannot walk within the scene in 360-degree videos, that is, 3D translation is not allowed besides the passive motion caused by a moving camera. Nevertheless, users can freely select their point of view during playback of 360-degree videos (i.e., 3D rotation is possible). Conveniently, a 360-degree video can be viewed via an ordinary web browser in that a user can pan around by clicking and dragging. Alternatively, a 360-degree video can be observed via a head-mounted display (HMD) in that a user can pan around simply by rotating his head.

In ordinary monoscopic videos, real-time tracking techniques rely on either offline datasets or online learning to train an appearance model, then apply the trained model to track potential targets frame by frame. However, tracking algorithms designed for ordinary videos may not perform well on 360-degree videos with their unique characteristics [3,4]. For example, occlusion problems are almost unavoidable in panoramic 360-degree videos. Moreover, an object may disappear from the left but reappear on the right border, or disappear from the top then reappear on the bottom border in 360-degree videos. Nonrigid deformation is obvious on 360-degree videos in an equirectangular format. Lighting and scale changes happen frequently in 360-degree videos due to continuous change of viewpoints. Camera shakiness is another common problem in aerial 360-degree videos captured by drones. Trackers designed for ordinary videos tend to confuse or even lose the tracking target under these circumstances in 360-degree videos. To this end, Cai et al. [5] adapted the Kernelized Correlation Filter (KCF) tracking algorithm to work on 360-degree videos. Delforouzi et al. [6] modified the Track-Learn-Detection (TLD) tracker to fit the needs of 360-degree videos. Nevertheless, a thorough evaluation of state-of-the-art tracking algorithms on 360-degree videos is still missing. For this purpose, we implement eight popular tracking techniques and evaluate their performance on 360-degree videos. To make a fair comparison of both quality and time of tracking, we adopt the default parameters of these trackers in an open-sourced library called OpenCV. The experimental results are analyzed to reveal the pros and cons of these tracking algorithms on 360-degree videos.

The contribution of this paper is a thorough evaluation of eight popular tracking algorithms on 360-degree videos. Both qualitative and quantitative comparisons are made in terms of accuracy and speed. According to the experimental results, we discuss the strengths and weaknesses of these trackers on 360-degree videos, and suggest potential ways to adapt them to 360-degree videos for better tracking performance. As a basis of the comparison, we capture nine 360-degree videos in a variety of scenarios. Three of them are captured on the ground and six of them are captured in the air.

Positions of interesting targets in these 360-degree videos are manually marked in each frame as the ground truth of tracking. The dataset containing these nine 360-degree videos with the ground truth is provided (online link in supplementary materials) to be a benchmark for future research.

2. Background

With the advance of virtual reality technology in the past few years, 360-degree images and videos have become a blooming research topic. Neng and Chambel [7] designed and evaluated 360-degree hypervideos that allow users to explore and navigate through links. Berning et al. [8] adopted 360-degree interactive video to create evaluation scenarios where users can select their point of view during playback. Rupp et al. [9] used 360-degree videos as a learning tool and analyzed the effects of immersiveness of three devices: a smartphone, a Google Cardboard, and an Oculus Rift. Pakkanen et al. [10] compared three interactive ways for 360-degree video playback: remote control, head orientation, and hand gesture. Huang et al. [11] presented an automatic approach to generate spatial audio for panorama images based on object detection and action recognition.

Giving the initial position of an unknown object, the purpose of tracking is to locate the object in successive frames of a video. Mousas [12] proposed a method for controlling motions of a virtual partner character based on a performance-capturing process using multiple inertial measurement units (IMUs). Instead of relying on IMUs for human tracking, this paper focuses on vision-based methods for unknown object tracking. Among all the existing online visual tracking algorithms, we choose eight modern and popular trackers for evaluation and comparison on 360-degree videos.

The Boosting tracker, proposed by Grabner et al. [13], is an online version of the AdaBoost feature selection algorithm. The online boosting algorithm maintains a global classifier pool of weak classifiers with multiple selectors. Each new training sample is used to update each weak classifier in the pool. A cascade system initializes the first selector with the current sample's importance, selects the best weak classifier with the least error, and passes the estimated importance to the next selector until all selectors have been updated. In the end, a strong classifier is chosen from the best weak classifiers, and the worst weak classifier is replaced with a random one. The Boosting tracker utilizes the initial target area in the current frame as a positive example, and exploits other areas with the same size around the target as negative examples. Then, the online-trained classifier searches the neighborhood for potential targets in the next frame. The Boosting tracker can handle temporary occlusions as well as complex backgrounds.

The Multiple Instance Learning (MIL) tracker, proposed by Babenko et al. [14], extends the online boosting algorithm by using a set of image patches (called a bag) instead of a single sample for training. A bag containing at least one positive example is called a positive bag, otherwise it is called a negative bag. The MIL tracker collects lots of small image patches centered at the tracking object as potential positive bags, and chooses the best one to be the positive example. This strategy not only prevents the MIL tracker from losing important information but also avoids the mislabeling problem.

The MedianFlow tracker, proposed by Kalal et al. [15], is a bidirectional approach that combines forward and backward tracking. The forward and backward consistency is analyzed as a quality measure to assist the tracking. The MedianFlow tracker constructs both forward and backward trajectories at each time instant, and their corresponding errors are estimated. The trajectory with the minimum forward-backward error is chosen as the candidate for the succeeding tracking. As a result, the MedianFlow tracker is more reliable to follow objects with consistent movement.

The Minimum Output Sum of Squared Error (MOSSE) tracker, proposed by Bolme et al. [16], is a tracker based on correlation filters. It achieves high efficiency by computing correlation in time domain. The MOSSE filter improves the ASEF filter to overcome the potential overfitting problem. The MOSSE tracker calculates the minimum output sum of square error to find out the most possible location of the tracking object. The benefits of using a correlation filter make the MOSSE tracker more robust to the problems of scaling, rotation, deformation, and occlusion compared to traditional approaches. Also,

MOSSE is more flexible than other correlation-filter-based trackers because the target is not required to be in the center of the image in the beginning of tracking.

The TLD tracker, proposed by Kalal et al. [17], is mainly composed of three parts: a tracker, a learner, and a detector. The job of the tracker is to follow the target through consecutive frames; the learner relies on a P-expert and an N-expert to estimate misdetection and false alarm, respectively, then updates the detector. The detector locates potential targets according to an appearance model, feeds the outputs to the learner, and corrects the tracker if necessary. The TLD tracker is well known for its ability of failure recovery at the expense of instability. Compared to other online trackers struggling with the problem of accumulating errors, the combination of tracking and detecting modules makes the TLD tracker more reliable for long-term tracking.

The KCF tracker, proposed by Henriques et al. [18], extends the MOSSE concept and takes advantage of overlapping regions in multiple positive samples. The abundant data is computed in Fourier domain to increase the learning speed. The KCF tracker emphasizes the importance of the negative samples and tends to use more samples for better training. To this end, a cyclic shift is applied to generate more samples from each important sample. The characteristic of circulant matrices for regression samples is utilized to speed up the computation. Also, kernel tricks are exploited to deal with the problem of nonlinear regression. Instead of scanning through raw pixels, the KCF tracker extracts the Histogram of Gradient (HoG) features to improve the accuracy of tracking.

The Generic Object Tracking Using Regression Networks (GOTURN) tracker, proposed by Held et al. [19], adopts an offline dataset to train a Convolutional Neural Network (CNN) model in advance. Then, it relies on the generated model for online tracking. The process of pretraining takes advantage of readily available information in offline datasets to learn both target appearance and motion relationship. Without the requirement to update CNN weights in run-time, the GOTURN tracker has another significant advantage of online tracking speed. Although it is not necessary to include specific tracking targets in the dataset for pretraining, the GOTURN tracker tends to favor objects in the training set over objects that are not in the training set. A potential issue of the GOTURN tracker is the quality of the pretrained model that can seriously affect the performance of the online tracking process.

The Channel and Spatial Reliability Tracker (CSRT), proposed by Lukezic et al. [20], is based on the Discriminative Correlation Filter (DCF) algorithm. It improves the DCF tracker by introducing spatial and channel reliability. The spatial reliability map is used to find out the optimal filter size. The ability to adjust filter size makes the CSRT tracker better than the traditional DCF algorithm by excluding unrealistic samples. Another benefit from the spatial reliability map is its ability to handle nonrectangular targets. The channel reliability is measured to weigh the importance of each channel filter, then combine them to get the final response map. Using only the HoGs and Colorname standard feature sets, the CSRT tracker can achieve an impressive accuracy with real-time speed.

3. Experiments

3.1. Experiment Setup

To measure the performance of eight trackers on 360-degree videos in a variety of situations, we prepared a dataset containing nine 360-degree videos captured using a Garmin Virb 360-degree camera. The 360-degree camera can be installed on top of a helmet for ground video capturing as shown in Figure 2a. Alternatively, the 360-degree camera can be attached to a drone for aerial video capturing as shown in Figure 2b. A Garmin Virb 360-degree camera contains two 12-megapixel sub-cameras that are opposite to each other. Each sub-camera has a wide field of view (FOV) of 202 degrees. At each time instant, the hardware inside the camera analyzes the overlap between two images captured by sub-cameras, then aligns and stitches two images together to form a seamless 360-degree image.



Figure 2. Two ways to set up an omnidirectional camera: (a) installed on top of a helmet for ground video capturing, (b) attached to a drone for aerial video capturing.

One problem of 360-degree videos is the huge file size. Typically, a 360-degree video contains 30 360-degree images per second, and each 360-degree image has a resolution of 3840×2160 with three color channels, resulting in a total of 746 MB per second. Video compression techniques can be applied to effectively reduce the size of the captured video file. Smaller video size can increase the frame rate of tracking and make the motion between two consecutive frames smaller, and hence improve the performance of tracking. On the other hand, high compression ratio and low bit rate can reduce the video quality, and hence degrade the accuracy of tracking. Wang et al. [21] studied the influences of the choice of video coding parameters on the performance of visual object tracking. In default setting, the hardware inside the Garmin Virb 360-degree camera applies the most commonly used AVC/H.264 encoding and generates a standard MP4 video file with a maximum bit rate of 120 Mbps. To make a fair comparison of eight trackers, our experiments are made based on the same video encoding and bit rate in all nine video sequences.

As shown in Table 1, these 360-degree video sequences cover multiple scenarios, each containing a combination of characteristics such as viewpoint change, occlusion, deformation, lighting change, scale change, and camera shakiness. Each 360-degree video sequence lasts 100~1000 frames with a resolution of 3840×2160 . For speedup purpose, all video sequences are down-sampled to 1920×1080 in our tracking experiments. The benchmark machine is a PC with 3.2 GHz CPU and 16 GB RAM. The operating system is Microsoft Windows 10.

Table 1. Our dataset containing nine 360-degree videos, each with a combination of special characteristics.










Sequences	Characteristics	Viewpoint	Occlusion	Deformation	Lighting	Scale	Shakiness
 Sequence 1		v			v		
 Sequence 2		v	v	v			

Table 1. Cont.

Sequences	Characteristics	Viewpoint	Occlusion	Deformation	Lighting	Scale	Shakiness
		v		v			
		v				v	
						v	
		v		v			
		v					
							v
					v		

3.2. Experiment Process

Our implementation of eight tracking algorithms was based on the open-sourced OPENCV library with Version 3.4.2 (Intel Corporation, Santa Clara, CA, USA). All eight trackers were initialized with default parameters. Among these trackers, the GOTURN tracker was the only one based on the Convolutional Neural Network (CNN) and utilized a standard Caffe model for online tracking. For each frame in a 360-degree video, the centroid of the tracking target was marked up manually in advance as the ground truth of tracking. Figure 3 shows the flowchart of the proposed experiment to evaluate eight trackers on nine 360-degree videos. Each tracker was executed on individual 360-degree video sequences in turn to measure the tracking speed in terms of Frames Per Second (FPS). A spatial displacement (in pixels) was computed as the absolute distance between the tracker’s output position and the ground truth. If the displacement was smaller than a predefined tolerated threshold, the frame was counted as a correct tracking frame. The accuracy was defined as the ratio of the number of correct tracking frames to the number of all frames. Because only the GOTURN, TLD, CSRT, and MedianFlow trackers could adjust and update target window size dynamically, adaptable window size was implemented in these four trackers for qualitative evaluation. To make a fair comparison of eight trackers, the size of the target window was not considered for quantitative evaluations. For the same reason, the Kalman filter was not applied for all trackers in our experiments.

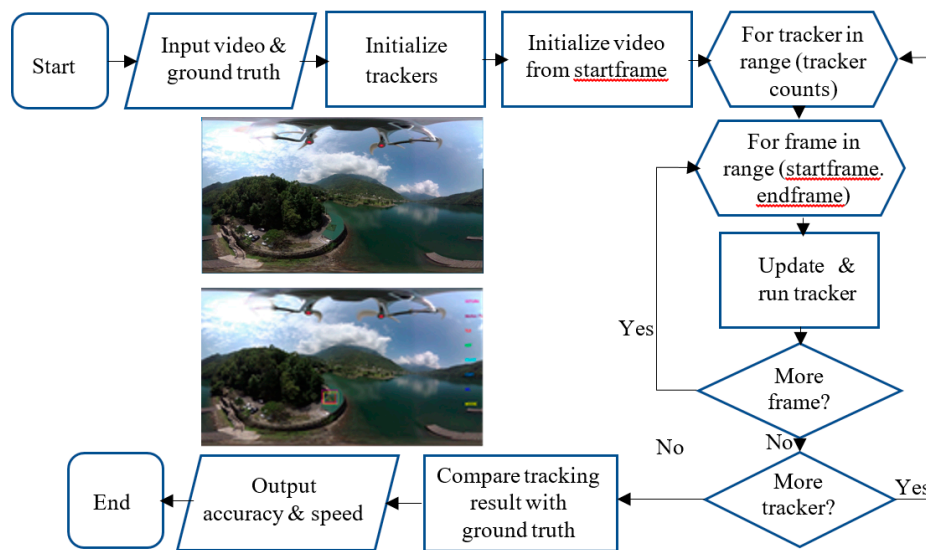


Figure 3. Flowchart of the proposed experiment to evaluate eight trackers on nine 360-degree videos.

3.3. Experiment Results

To demonstrate qualitative tracking outputs of eight trackers on nine 360-degree video sequences, six representative snapshots with a fixed interval of time in each 360-degree video sequence are shown in detail in Figures 4–12. The tracking results of all trackers are marked as rectangular windows with different colors in each snapshot (MOSSE:yellow; MIL:blue; MedianFlow:purple; BOOST:turquoise; TLD:red; KCF:green; GOTURN:pink; CSRT:cyan). The video sequence 1 was captured by a moving biker. The tracking target was another bike with huge scale change, some viewpoint change, and minor lighting change over time. Among eight trackers, the CSRT, MIL, BOOST, and MOSSE trackers performed quite well, but other trackers lost the target in the middle of the sequence as shown in Figure 4. The video sequence 2 was captured by a moving motorcycle. The tracking target was a stadium on one side of the road. The building was occasionally occluded by trees and streetlamps. All trackers were affected by the occlusion problem and so only produced decent tracking results as shown in Figure 5. Especially, the MedianFlow tracker confused the tracking target with other obstacles. It suffered seriously from temporal occlusion in this sequence. The video sequence 3 was captured by a drone with an overlooking view of a lake. The tracking target was the roof of a green building. The shape of the target deformed dramatically due to the nature of 360-degree videos in an equirectangular format. Luckily, some trackers could still follow the target smoothly for a short period of time.



Figure 4. Tracking results on 360-degree video sequence 1 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

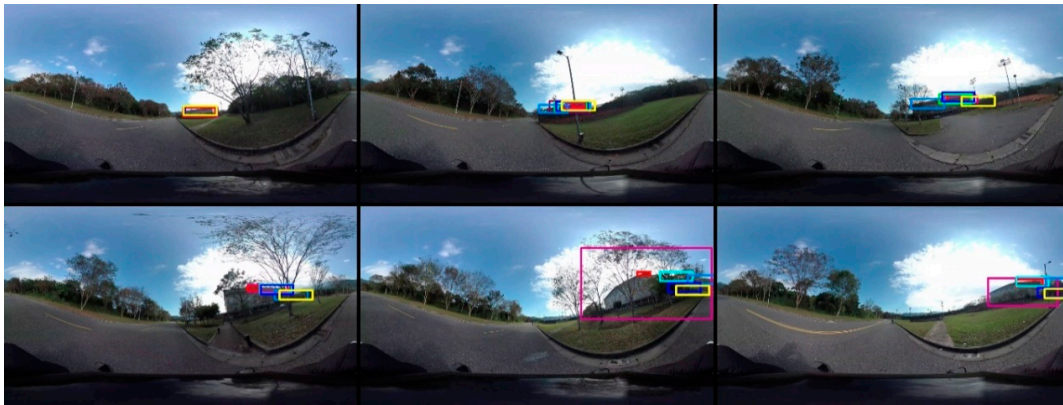


Figure 5. Tracking results on 360-degree video sequence 2 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

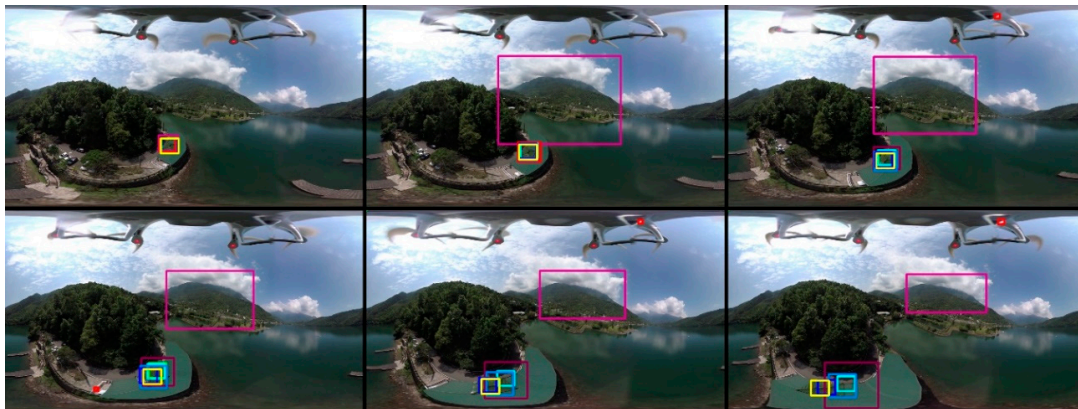


Figure 6. Tracking results on 360-degree video sequence 3 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

The video sequence 4 was captured by a drone flying around a lake harbor. The tracking target was a small boat docking at a pier. The scale of the target changed a lot over time as shown in Figure 7. Though the KCF and MOSSE trackers achieved a fair accuracy at the beginning, they lost the target in the middle of the sequence. Even worse, the GOTURN and TLD trackers got confused and tracked the wrong objects in the early stage. The video sequence 5 was captured by a drone flying along a lakeshore. The tracking target was a lakeside building with a red roof. The scale of the building changed slowly over time, hence the MedianFlow tracker performed quite well. The sequence 5 contained another nature of 360-degree videos in that the tracking target disappeared from one side and reappeared on the other side of the panoramic image. Most trackers cannot recover from this problem. Interestingly, the TLD tracker correctly recovered the target as shown in the last snapshot of Figure 8. The video sequence 6 was captured by a drone flying across a lake. The tracking target was a fast-moving boat with apparent viewpoint change. With the problem of large motion in this sequence, only the GOTURN tracker obtained great results. In comparison, the KCF and MOSSE trackers lost the tracking target very early as shown in Figure 9.

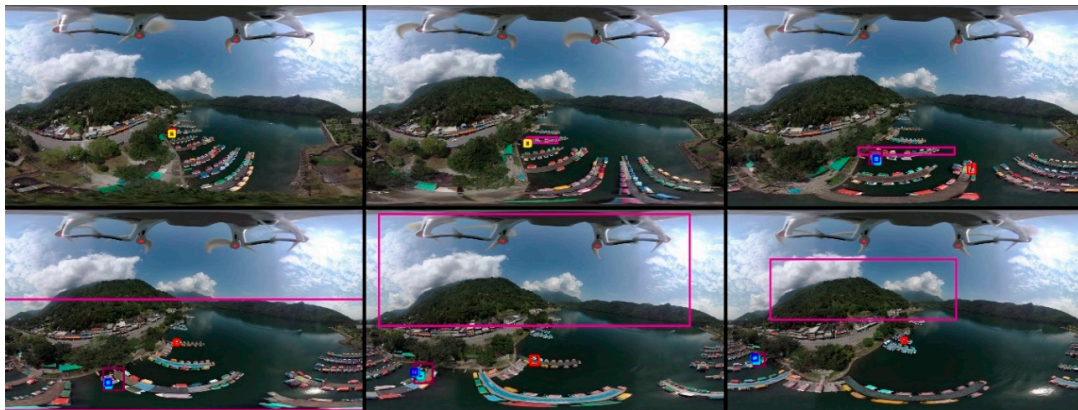


Figure 7. Tracking results on 360-degree video sequence 4 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

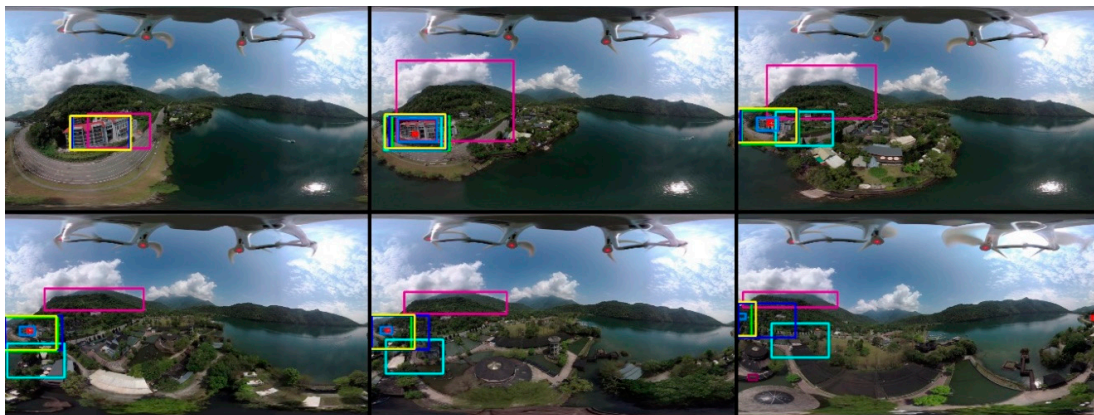


Figure 8. Tracking results on 360-degree video sequence 5 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

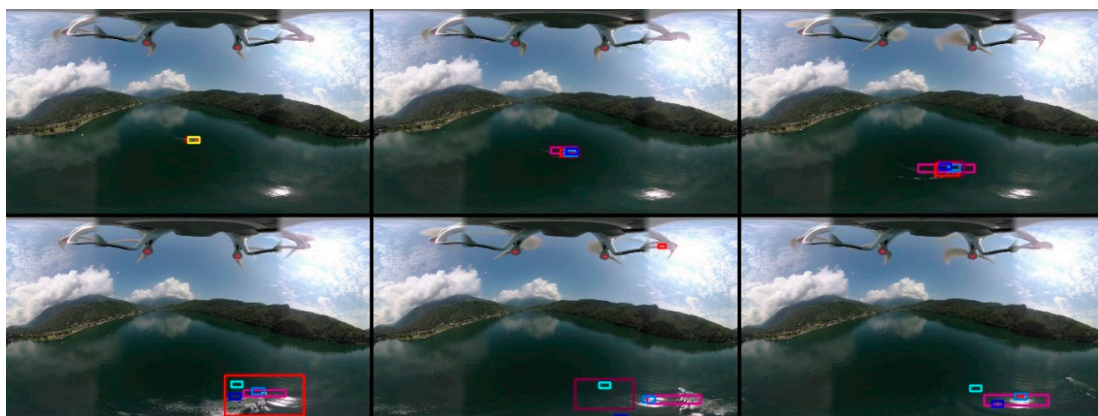


Figure 9. Tracking results on 360-degree video sequence 6 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

The video sequence 7 was captured by a moving biker. The tracking target was a large building with slow viewpoint change, and some partial occlusion. Although several trackers received decent scores, they were not really focused on the center of the building. Only the BOOST tracker accurately tracked the whole building throughout this sequence as shown in Figure 10. The video sequence 8 was captured by a drone flying on windy days. The characteristic of this sequence was camera shakiness which is a common problem on drone-recorded 360-degree videos. Even with the jittery

motion and target deformation caused by the shaking camera, most trackers still performed quite well throughout the sequence as shown in Figure 11. The video sequence 9 was captured by a drone flying along a seashore. The tracking target was the summit of a mountain, and the illumination changed dramatically over space and time. In the middle of the sequence, the sun sat just behind the mountain top. The problem of backlighting caused tracking loss for the KCF tracker, and tracking error for the TLD and GOTURN trackers. Surprisingly, other trackers still followed the target very well as shown in Figure 12. In summary, Figure 13 outlines the quantitative results of eight trackers on nine sequences. The vertical axis indicates the tracking accuracy, and the horizontal axis represents the predefined value of the tolerated threshold.

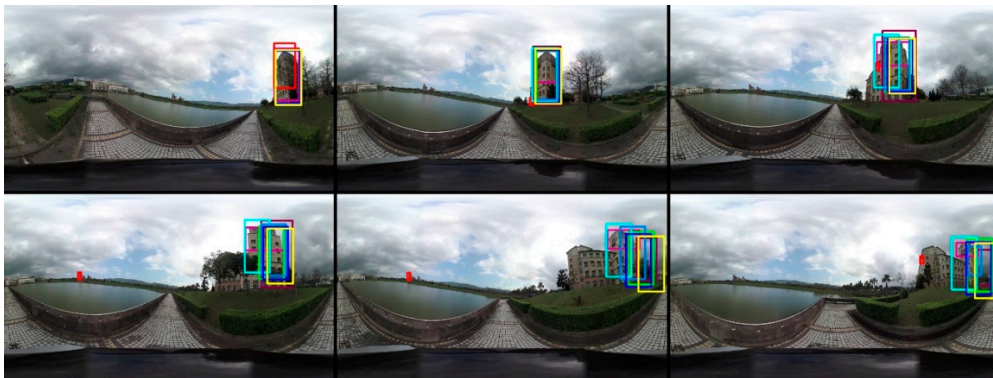


Figure 10. Tracking results on 360-degree video sequence 7 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).



Figure 11. Tracking results on 360-degree video sequence 8 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

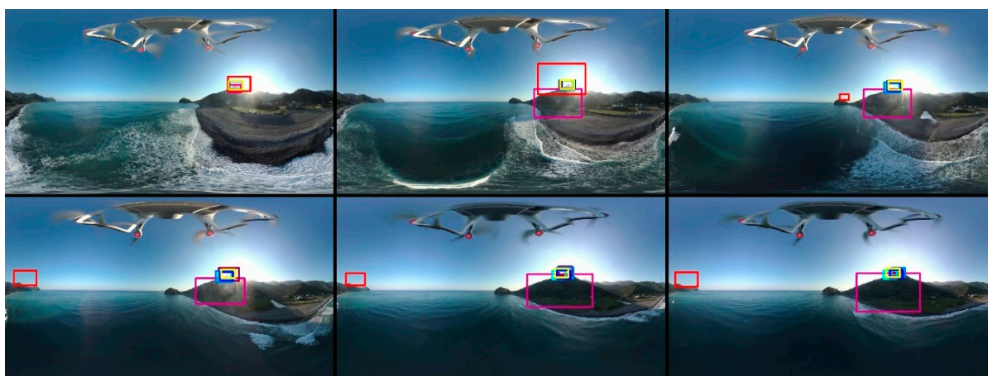


Figure 12. Tracking results on 360-degree video sequence 9 (tracking windows with different colors: MOSSE MIL MedianFlow BOOST TLD KCF GOTURN CSRT).

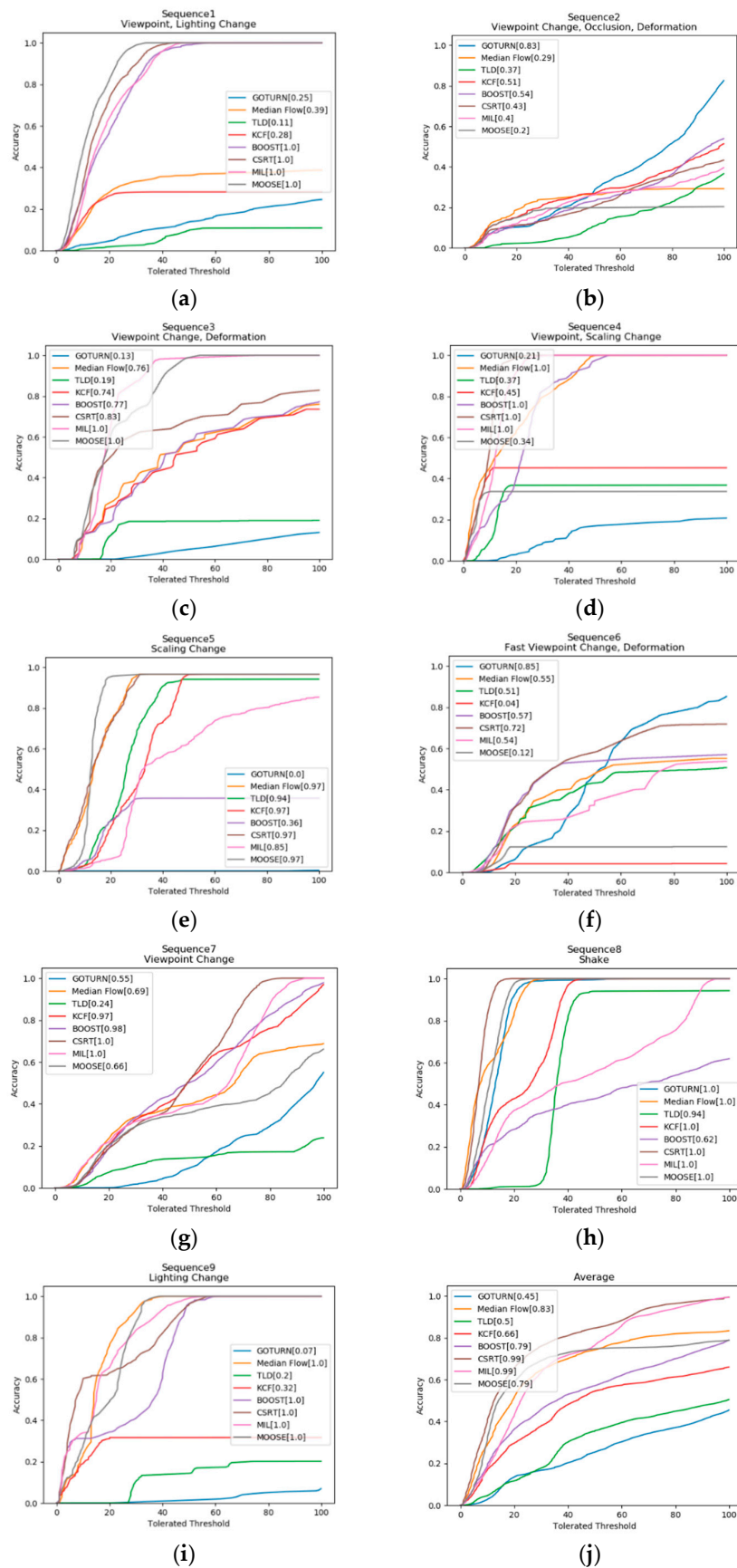


Figure 13. Accuracy comparison of eight trackers on nine 360-degree video sequences. (a) Seq.1, (b) Seq.2, (c) Seq.3, (d) Seq.4, (e) Seq.5, (f) Seq.6, (g) Seq.7, (h) Seq.8, (i) Seq.9, (j) Overall average.

4. Discussion

In terms of Frames Per Second (FPS), Table 2 summarizes the speed comparison of all eight trackers in nine video sequences. According to the experimental results, the MOSSE is the fastest tracker with an average of 3776 FPS. The KCF is the next efficient tracker with an average of 175 FPS. The MedianFlow tracker can also achieve 63 FPS. Among all eight trackers, The TLD is the slowest tracker. In fact, it is about 600 times slower than the MOSSE tracker and not suitable for real-time applications.

Table 2. Speed comparison of eight trackers on nine video sequences in terms of Frames Per Second (FPS).

Trackers Sequences	GOTURN	MedianFlow	TLD	KCF	BOOST	CSRT	MIL	MOSSE
Sequence 1	23.7	62.8	2.6	188.9	41.6	38.9	12.8	4118.2
Sequence 2	20.5	66.0	9.5	204.8	35.8	35.6	14.3	2250.0
Sequence 3	16.0	61.8	2.8	172.9	26.7	32.8	14.3	2532.2
Sequence 4	14.3	62.8	4.9	309.5	56.9	47.6	14.9	10356.8
Sequence 5	16.4	61.4	5.6	39.0	22.7	34.8	13.0	457.0
Sequence 6	21.3	65.2	7.1	304.8	49.8	47.3	17.2	8614.0
Sequence 7	21.2	63.1	8.7	41.4	15.8	22.8	12.1	440.3
Sequence 8	23.4	63.6	8.0	104.3	23.2	31.5	12.1	1233.8
Sequence 9	19.3	64.1	4.9	210.8	37.4	37.8	12.7	3982.4
Average	19.6	63.4	6.0	175.1	34.4	36.6	13.7	3776.1

In terms of tracking quality, Table 3 summarizes the accuracy comparison of all eight trackers in nine video sequences. The strengths and weaknesses of all eight trackers are outlined in Table 4. The GOTURN is the only tracker based on deep learning but does not perform well in terms of accuracy on 360-degree videos. Interestingly, it works well in some special cases. For example, it is the only tracker that could flawlessly track a fast-moving boat in sequence 6, possibly because the pretrained dataset contained boats. In fact, the performance of GOTURN heavily depends on the appearance model of the tracking target. Hence, we believe the GOTURN tracker can be improved by training specific target models for 360-degree videos in advance.

Table 3. Overall accuracy comparison of eight trackers on nine video sequences (bad: 0~20%; poor: 20~40%; ok: 40~60%; good: 60~80%; great: 80~100%).

Trackers Sequences	GOTURN	MedianFlow	TLD	KCF	BOOST	CSRT	MIL	MOSSE
Sequence 1	poor	poor	bad	poor	great	great	great	great
Sequence 2	great	poor	poor	ok	ok	ok	poor	bad
Sequence 3	bad	good	bad	good	good	great	great	great
Sequence 4	poor	great	poor	ok	great	great	great	poor
Sequence 5	bad	great	great	great	poor	great	great	great
Sequence 6	great	ok	ok	bad	ok	good	ok	bad
Sequence 7	ok	good	poor	great	great	great	great	good
Sequence 8	great	great	great	great	good	great	great	great
Sequence 9	bad	great	bad	poor	great	great	great	great

Table 4. Strengths and weaknesses of all eight trackers on 360-degree videos.

Tracker	Features	Principle	Strength	Weakness	Improvement Suggestion
GOTURN		Pretrained CNN model	Recovery from failure and occlusion	Target not in training data	Include specific targets for training in advance
MedianFlow		Min forward–backward error	Reliable on slow changing target	Fast-moving target	Support motion detection
TLD		Track, learn, and detect	Recovery from failure and occlusion	High false alarm	Combine with reliable filter
KCF		Kernelized correlation filter	Report tracking failure	Fixed target size	Adaptable target size
BOOST		AdaBoost	Decent accuracy	Seldom report tracking failure	Adaptable tolerance
CSRT		Discriminative correlation filter	Robust and high accuracy	Long-term occlusion	Failure recovery with spatial relationship
MIL		Multi-instance learning	High accuracy	Long-term occlusion	Failure recovery with spatial relationship
MOSSE		Min square error	High tracking speed	Fixed target size	Adaptable target size

Generally, the MedianFlow tracker performs well on consistent and slowly changing video sequences. However, occasional occlusion prevents it from making an agreement in bidirectional analysis and the tracking fails as shown in video sequence 2.

The TLD is a slow tracker with high false detect rate but works well in the case of failure recovery. Especially, the TLD tracker is a good choice to track a target that disappears from one place and reappears in another place in 360-degree videos.

The KCF tracker performs well on ordinary videos but not on 360-degree videos due to its fix-sized filters. The characteristics of 360-degree videos such as scale change, viewpoint change, and deformation easily lead the KCF tracker to a track loss. Thus, it is only useable to track plain targets that do not contain these characteristics.

The Boost tracker achieves a fair accuracy on 360-degree videos, though it does not sense tracking failure and continues to track a wrong target as shown in video sequence 5. The parameters of tolerance need to be adjusted accordingly to avoid false tracks for the Boost tracker.

The CSRT tracker is a good choice for tracking on 360-degree videos because it detects target objects using the HoG features instead of raw pixels. It can adjust the size of target window dynamically as well. Nonetheless, it still has a hard time recovering from a temporarily disappearing target as shown in video sequence 2, or tracking a fast-moving target as shown in video sequence 6.

The MIL tracker can properly handle most of the cases on 360-degree videos. Its weakness is the problem of occlusion caused by change of viewpoints. The MIL tracker tends to fail in recovering the tracking target even after the occlusion.

For applications with high-speed demand or large motion, the MOSSE tracker is the best choice since its tracking speed is significantly higher than other trackers, though the fix-sized tracking window could be a problem for video sequences with huge scale change.

A typical example of 360-degree videos in an equirectangular format is shown in Figure 14. An ideal tracker should be able to tackle the problems in 360-degree videos such as viewpoint change, occlusion, deformation, lighting change, scale change, and camera shakiness. For viewpoint change caused by a moving camera, a motion model is helpful to assist tracking. To handle occlusion problems, the ability to recover the temporarily missing target is essential. To alleviate the nonrigid deformation,

trackers must learn and update the appearance model in run-time. To accommodate lighting change, extracting illumination-robust features is critical. To solve the problem of scale change, adaptable and dynamic target window size is beneficial. To deal with a shaking camera, trackers should measure and compensate the global motion.



Figure 14. A typical example of 360-degree videos in an equirectangular format with several characteristics: seashore deformation, island scale change, building occlusion, mountain summit lighting change.

An inherent problem of 360-degree videos in an equirectangular format is the image distortion, especially for the northern and southern polar areas in a panoramic image. The distortion problem affects the performance of tracking in two aspects. First, the motion of the tracking target is distorted after an equirectangular projection [22]. In video sequence 6, a boat moving in a straight line looks like it is moving in a curve in the 360-degree video in an equirectangular format. The distortion of trajectory degrades the accuracy of all trackers, especially for fast-moving targets. Nevertheless, the GOTURN tracker can handle this problem properly as long as it includes the training samples with distorted motions in the process of pretraining. Second, the tracking target suffers from nonrigid deformations in an equirectangular format. In video sequence 2, the deformation of the target building makes straight lines become curves. Thus, it tends to cause a track loss for trackers using a static target appearance model. Surprisingly, most trackers survive the slow target deformation in this case except the TLD tracker. The deformed target triggers frequent reinitialization in the TLD modules, and the target appearance is prone to drift in the presence of occasional occlusions. In video sequence 3, the tracking target is accompanied by both motion distortion and target deformation. As a result, most trackers are unstable and achieve low tracking accuracy in this case.

5. Conclusions

The problems of viewpoint change, occlusion, deformation, lighting change, scaling change, and shakiness occur frequently in 360-degree videos. According to our experiments with maximum tolerated threshold, the CSRT achieves the best overall accuracy and is the most robust tracker on 360-degree videos. Alternatively, the MOSSE is the most efficient tracker in terms of speed. For future work, an ideal tracker that can deal with these problems in 360-degree videos is crucial. We believe that a multimodal fusion is beneficial in combining abilities of failure recovery, robustness, and adaptable

target size for online tracking on 360-degree videos. A Kalman filter can also be applied for better prediction and stabilization of unknown object tracking in 360-degree videos. Our dataset containing nine 360-degree videos with ground truth is accessible through the link at the end of the paper. It can be utilized as a benchmark for future research.

Supplementary Materials: The dataset is available online at: <https://drive.google.com/open?id=1Ybp0G6yWXYCsP06nzEMRJR-exK0DSos8>.

Author Contributions: Conceptualization, M.-T.Y.; Methodology, T.-W.M.; Software, T.-W.M.; Validation, T.-W.M.; Formal analysis, T.-W.M.; Investigation, M.-T.Y.; Resources, M.-T.Y.; Data curation, T.-W.M.; Writing—original draft preparation, T.-W.M.; Writing—review and editing, M.-T.Y.; Visualization, T.-W.M.; Supervision, M.-T.Y.; Project administration, M.-T.Y.; Funding acquisition, M.-T.Y.

Acknowledgments: This research is partially funded by the Ministry of Science and Technology (MOST) under contracts 107-2221-E-259-019-MY2, 107-2511-H-259-004-MY2, 108-2218-E-259-001, 108-2634-F-259-001 through Pervasive Artificial Intelligence Research (PAIR) Labs, Taiwan.

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

References

1. Zia, O.; Kim, J.; Han, K.; Lee, J.W. 360° Panorama Generation using Drone Mounted Fisheye Cameras. In Proceedings of the IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 11–13 January 2019; pp. 1–3.
2. Kelling, C.; Väättäjä, H.; Kauhanen, O. Impact of device, context of use, and content on viewing experience of 360-degree tourism video. In Proceedings of the International Conference on Mobile and Ubiquitous Multimedia, New York, NY, USA, 26–29 November 2017.
3. Afzal, S.; Chen, J.; Ramakrishnan, K.K. Characterization of 360-degree Videos. In Proceedings of the ACM SIGCOMM 2017 Workshop on Virtual Reality and Augmented Reality Network, Los Angeles, CA, USA, 25 August 2017.
4. Boonsuk, W. Evaluation of Desktop Interface Displays for 360-Degree Video. Master's Thesis, Iowa State University, Ames, IA, USA, 2011.
5. Cai, C.; Liang, X.; Wang, B.; Cui, Y.; Yan, Y. A Target Tracking Method Based on KCF for Omnidirectional Vision. In Proceedings of the 37th Chinese Control Conference, Wuhan, China, 25–27 July 2018; pp. 2674–2679.
6. Delforouzi, A.; Tabatabaei, S.; Shirahama, K.; Grzegorzec, M. Unknown object tracking in 360-degree camera images. In Proceedings of the International Conference on Pattern Recognition, Cancun, Mexico, 4–8 December 2016.
7. Neng, L.; Chambel, T. Get around 360 hypervideo: Its design and evaluation. *Int. J. Ambient Comput. Intell.* **2012**, *4*, 40–57. [[CrossRef](#)]
8. Berning, M.; Yonezawa, T.; Riedel, T.; Nakazawa, J.; Beigl, M.; Tokuda, H. pARnorama: 360 degree interactive video for augmented reality prototyping. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 8–12 September 2013; pp. 1471–1474.
9. Rupp, M.; Kozachuk, J.; Michaelis, J.; Odette, K.; Smither, J.; McConnell, D. The effects of immersiveness and future VR expectations on subjective-experiences during an educational 360 video. In Proceedings of the International Annual Meeting of the Human Factors and Ergonomics Society, Washington, DC, USA, 19–23 September 2016; pp. 2108–2112.
10. Pakkanen, T.; Hakulinen, J.; Jokela, T.; Rakkolainen, I.; Kangas, J.; Piippo, P.; Raisamo, R.; Salmimaa, M. Interaction with WebVR 360 video player: Comparing three interaction paradigms. In Proceedings of the IEEE Virtual Reality, Los Angeles, CA, USA, 18–22 March 2017; pp. 279–280.
11. Huang, H.; Solah, M.; Li, D.; Yu, L. Audible Panorama: Automatic Spatial Audio Generation for Panorama Imagery. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019.
12. Mousas, C. Performance-driven dance motion control of a virtual partner character. In Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces, Reutlingen, Germany, 18–22 March 2018; pp. 57–64.
13. Grabner, H.; Grabner, M.; Bischof, H. Real-Time Tracking via On-line Boosting. In Proceedings of the British Machine Vision Conference, Edinburgh, UK, 4–7 September 2006.

14. Babenko, B.; Yang, M.; Belongie, S.J. Visual tracking with online Multiple Instance Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 983–990.
15. Kalal, Z.; Mikolajczyk, K.; Matas, J. Forward-Backward Error: Automatic Detection of Tracking Failures. In Proceedings of the International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2756–2759.
16. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
17. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. [[CrossRef](#)] [[PubMed](#)]
18. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [[CrossRef](#)] [[PubMed](#)]
19. Held, D.; Thrun, S.; Savarese, S. Learning to Track at 100 FPS with Deep Regression Networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
20. Lukezic, A.; Vojír, T.; Zajc, L.C.; Matas, J.; Kristan, M. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4847–4856.
21. Wang, P.; Zhang, L.; Wu, Z.; Xu, R. Research on video coding parameters affecting object tracking. In Proceedings of the International Conference on Wireless Communications & Signal Processing, Nanjing, China, 15–17 October 2015.
22. Sacht, L.; Carvalho, P.; Velho, L.; Gattass, M. Face and Straight Line Detection in Equirectangular Images. In Proceedings of the Workshop de Visão Computacional, São Paulo, Brazil, 4–7 July 2010.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).