# Evaluation of Visual Object Trackers on Equirectangular Panorama

Ugur Kart[1], Joni-Kristian Kämäräinen[1] Lixin Fan[2] and Moncef Gabbouj, Fellow IEEE[1]
[1]*Department of Signal Processing, Tampere University of Technology, 33720, Tampere, Finland*
[2]*Nokia Technologies, 33100, Tampere, Finland*
{*ugur.kart, joni.kamarainen, moncef.gabbouj*}*@tut.fi, lixin.fan@nokia.com*

Keywords: Tracking, Equirectangular, 360°-video

Abstract: Equirectangular (360° spherical) panorama is the most widely adopted format to store and broadcast virtual reality (VR) videos. Equirectangular projection provides a new challenge to adapt existing computer vision methods for the novel input type. In this work, we introduce a new dataset which consists of high quality equirectangular videos captured using a high-end VR camera (Nokia OZO). We also provide the original wide angle (8× 195°) videos and densely annotated bounding boxes for evaluating object detectors and trackers. In this work, we introduce the dataset, compare state-of-the-art trackers for object tracking in equirectangular panorama and report detailed analysis of the failure cases which reveal potential factors to improve the existing visual object trackers for the new type of input.

## 1 INTRODUCTION

Virtual Reality(VR) and 360° video have recently created a big disruption in multiple industries. Thanks to its ability to create immersive experiences, 360° VR has shown a big potential in areas such as education, entertainment and communication. Despite the idea of VR and 360° video have been around for a long time, their widespread usage is a rather new phenomenon. Hence, suitable content for research purposes is still largely missing as capturing devices have been commercially available only for a short duration.

A fundamental problem of computer vision is visual object tracking. From its subsets, this paper's scope covers model-free, single object tracking where the object's coordinates are fed into the system as a bounding box at the beginning of the sequence and the algorithm tries to track the object in the consecutive frames automatically. Due to its importance in various real-life applications such as autonomous driving systems, surveillance systems, medical imaging, robotics, human-computer interfaces, there is continuous interest on this topic (Wu et al., ; Held et al., 2016; Kalal et al., 2011; Hare et al., 2016; Kristan et al., 2016b).

In order to evaluate the performance of tracking algorithms systematically, many valuable benchmarks have been proposed in the literature. Among these, PETS (Ferryman and Ellis, 2010), OTB50 (Wu et al., ), ALOV300+ (Smeulders et al., 2014), VOT2014 (Kristan et al., 2014), VOT2015 (Kristan et al., 2015) and VOT2016 (Kristan et al., 2016a) are the few prominent and recent ones. However, they all focus on videos recorded with regular monoscopic cameras which have distinctly different characteristics as compared to equirectangular panorama 360° videos. Therefore, we believe that creating an annotated dataset designed with this purpose in mind is necessary and it was our main motivation in this work.

**Contributions –** We make the following novel contributions:

- We introduce a novel 360° visual object tracking benchmark dataset that will be made publicly available with its fisheye source videos, calibration data and ground truth for equirectangular panorama videos.
- We evaluate a set of state-of-the-art trackers on this novel dataset.
- We provide a detailed analysis of the common failure cases that reveals potential ways to improve the trackers for object tracking in equirectangular panorama videos.

## 2 Related work

**Equirectangular Panorama –** Oculus Rift type Virtual Reality (VR) Head-Mounted Displays (HMDs) are becoming popular and many affordable devices

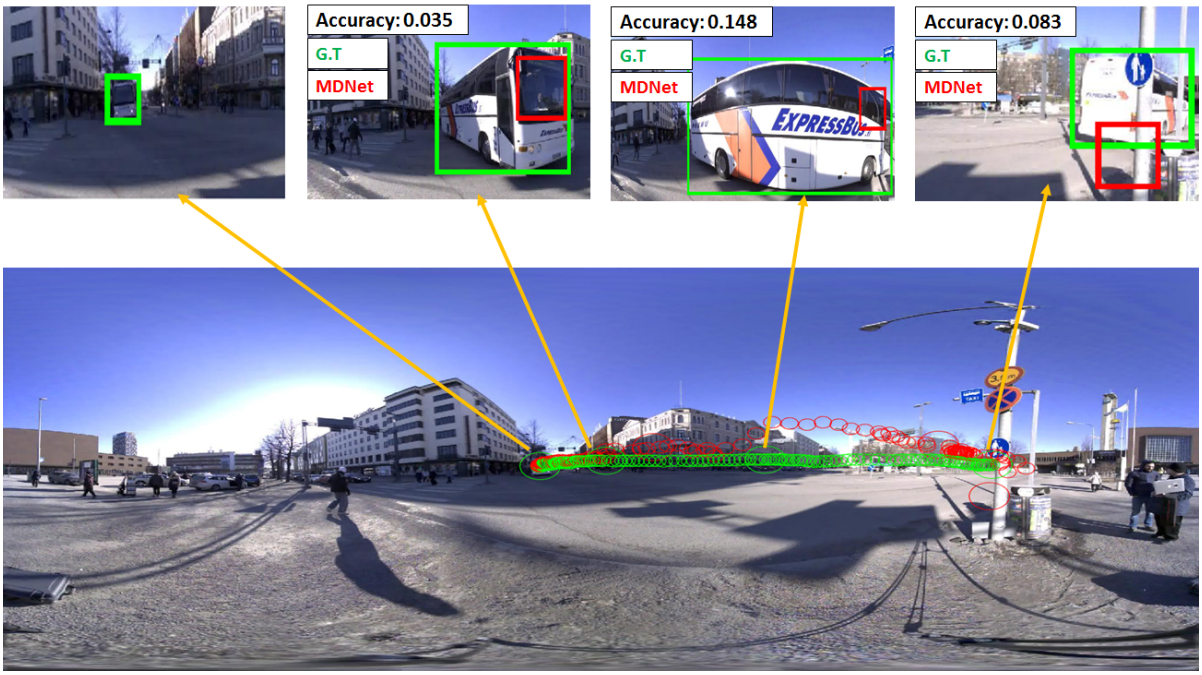


Figure 1: Equirectangular panorama projection creates large appearance changes (scale, geometric distortion and 3D view point) even for objects moving along a straight line trajectory. This will provide a novel challenge for existing trackers as shown in our experiments. In the above example the VOT2015 winner tracker misses the track of a bus steadily driving along the nearby street due to a dramatic appearance change.

are available for customers. Immersive experience requires ultra high resolution 360° videos which can be captured with high-end multi-camera devices such as Nokia OZO. Proper viewing requires transmission of 360° spherical frames and the equirectangular panorama has become the de facto standard. Equirectangular panorama is a well-known projection to map a 3D sphere (e.g., the world) to a 2D plane (equidistant cylindrical projection, e.g., a world map). Equirectangular panorama raises several technical challenges which have been recently investigated such as distorting projections (Carroll et al., 2009) and video encoding (Corbillon et al., 2017).

Hitherto there have been only a few attempts in vision community to process 360° content. Xiao et al. (Xiao et al., ) introduced a method for scene recognition from a panoramic view. They also introduced a new database of 360° panoramic images from 26 place categories on which they proposed an algorithm for classifying the place that the picture is taken from (e.g. a theater) and what direction the observer looks towards. However, they only studied still images. There are also recent attempts to convert 360° video to traditional 2D displays (Su et al., 2016; Su and Grauman, 2017). To the authors' best knowledge our work is the first to study visual object tracking in equirectangular panorama projection of spherical video. In their recent work, Cehovin et al. proposed a 360° video dataset to investigate the effects of apparent motion on visual tracking algorithms (Cehovin et al., 2016), but they use 360° video only to generate apparent motion patterns and the tracking itself is performed on traditional views generated by a suitable viewport projection. In our dataset, we also provide the original and unprocessed OZO fisheye source videos with the calibration data to facilitate future work where different projection schemes can be investigated for 360° video tracking. Our dataset is also larger than theirs (24k vs. 17k annotated frames, respectively).

**Visual Object Tracking –** Our problem is single object causal tracking without object re-identification (re-identification can however be investigated with our dataset as many objects frequently disappear and reappear in the scenes). In our experiments, trackers are initialized using the ground truth axis-aligned bounding boxes and trackers do not use prior knowledge about the targets. Such single camera, single object visual object tracking has a long history due to its importance in many vision problems. The field is expanding with growing number of sophisticated methods (Nam and Han, ; Danelljan et al., 2016) whereas some simpler methods might also work surprisingly well in a generic setting (Henriques et al., 2014). Trackers can be classified under two main families which are generative and discriminative ap-

proaches. Generative approaches create an appearance model of an object in the first frame and then try to find the best match in the next frames with a similarity metric. On the other hand, discriminative approaches, also known as tracking-by-detection, decide whether the object exists in a given patch in the next frame or not. Regardless of the family they belong, the fundamental idea underlying the visual object tracking is very similar in all and can be examined in five steps (Smeulders et al., 2014): *1) Target Region Representation* - A target region can be a bounding box, contours or multiple patches for deformable models but the clear trend is towards utilizing bounding boxes (Kristan et al., 2015; Kristan et al., 2016a); *2) Appearance Representation* - A set of visual features that represent the appearance of an image window varying from grayscale intensity values (Hare et al., 2016) to CNN features (Danelljan et al., 2016); *3) Motion Representation* - defines the mechanism for a candidate search region in the next frame and naturally, motion models which are centered at previous location are popular (Hare et al., 2016; Nam and Han, ); *4) Method* - It is the heart of the tracking algorithm and defines how it actually tracks the objects over the frames. Although searching without any prior spatial assumptions has its own strengths, it relies on appearance model heavily which tends to be erroneous due to the appearance changes during the video. Therefore, a particle filter is preferred in the presence of high computation power. *5) Model Updating* - Since the appearance of an object might change frequently in a sequence (view point changes), many modern trackers store last seen appearances to obtain a more comprehensive appearance database(Nam and Han, ; Danelljan et al., 2016; Kalal et al., 2011).

# 3 Equirectangular Panorama Video Dataset

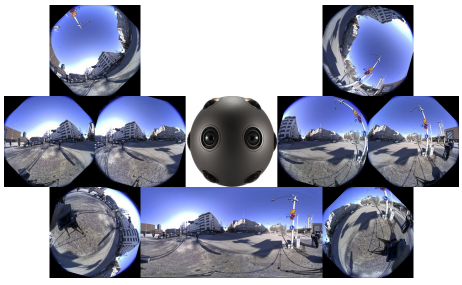

Figure 2: Unprocessed frames from Nokia OZO wide angle (195°) base lenses and a equirectangular panorama frame where the lenses are projected and stitched. Our dataset provides the stitched panorama and individual lens videos with their calibration data.

There is an increasing amount of spherical panorama videos in YouTube, but most of them are short and poor quality. Our motivation is to study vision methods for professional quality 360 °videos which are the main format to be used in future commercial television broadcasts and films. Our device of choice was Nokia OZO VR camera which is a high-end product aimed for professionals. We captured long videos from various academic events (lectures and scientific presentations, academic year opening ceremony and a department's Christmas party) and one video from the local train station. OZO records eight separate videos using synchronized fish-eye lenses with $2K \times 2K$ sensors and 195° angle of view wide angle lenses (Figure 2). We provide both the wide angle videos along with their factory calibration parameters and also generated equirectangular panorama videos made with a publicly available software for OZO (OZO Creator) that uses the same calibration parameters.

Table 1: Description of the annotated benchmark data. $N$ is the number of frames in the sequence, $N_{valid}$ is the total number of visible annotations. Please note that a frame can have multiple annotations for different objects which is the reason of having $N_{valid} \gg N$.

| *Video* | *Seq No* | *Object* | *#* | $N$ | $N_{valid}$ |
|---|---|---|---|---|---|
| Party | 1 | People | 13 | 3600 | 24841 |
| | 2 | Food Cart | 1 | 7605 | 5662 |
| Train station | 1 | Bus | 3 | 3600 | 4388 |
| | | Car | 3 | | |
| | | Minivan | 3 | | |
| | | People | 2 | | |
| Lecture | 1 | People | 1 | 3600 | 3530 |

For evaluating model-free visual trackers, we manually selected suitable sequences from videos and annotated various class instances for each sequence (see Table 1 for details). The Christmas party sequences contain more than 50 people, poor night time illumination, a lot of occlusions and many object distortions due to equirectangular projection and therefore is a challenging dataset for people tracking. 13 people were annotated and one food cart in a separate sequence. The Lecture video contains mainly the lecturer as a moving object, but provides a challenge for long-time tracking evaluation. The lecturer is frequently moving and being close to camera yields fast and severe scale changes and distortions. Due to the front spot lights the illumination is considerably poor towards the panorama image edges and that with partial occlusions may confuse trackers. *Train Station* video was captured outside on a sunny day. It was recorded in the front of a train station to include many vehicles and pedestrians passing by. There are occlusions due to other vehicles and pedestrians and

for cars passing through the nearby street there is significant view point and scale changes.

The manual annotations to provide the ground truth for the evaluated trackers was made by using the publicly available Vatic software. The software provides easy and fast to use GUI to annotate bounding boxes with objects' apparent occlusion and visibility properties. All annotations were made manually and object was marked "occluded" if more than 50% was missing and "not visible" if it was not possible to see the object.

# 4 Tracker Evaluation

In this section we report the results from quantitative and qualitative evaluation of four state-of-the-art and one baseline tracker with the equirectangular 360° video dataset introduced in Section 3. The overall workflow corresponds to the popular supervised evaluation mode protocol used in the annual VOT competitions (see (Kristan et al., 2016b) for more details).

## 4.1 Trackers

We selected the top performers of the recent Visual Object Tracking (VOT) Challenges: MDNet is the winning submission in VOT2015 and C-COT on pair with the winning submission (TCNN(Nam and Han, )) in VOT2016. These both use features extracted by convolutional neural networks (CNNs). MDNet and C-COT are below the real-time performance and therefore we included also the recent CNN-based tracker by Held et al. (Held et al., 2016) who provide super real-time (100fps) implementation. For all CNN-based trackers we used the implementations by the original authors and with their default parameters. As a baseline tracker we selected the OpenCV implementation of the original TLD tracker which is still popular among developers. As a more up-to-date algorithm, STRUCK tracker by Hare et al. (Hare et al., 2016) was also included (authors' public implementation with default parameter values). Properties of the selected trackers are summarized in Table 2 and below we briefly describe each tracker.

**C-COT –** C-COT achieved on pair performance to the winning method of the VOT2016 competition (Kristan et al., 2016a). Following the recent developments in the field, it uses MatConvNet (Vedaldi and Lenc, 2015) to extract features from patch locations. These features are then interpolated into continuous domain to achieve per pixel accuracy. The interpolated features are used to learn a linear convolution operator which maps a given patch $x$ to a confidence score $s$ which is then utilized for finding the image patch with maximum score.

**MDNet –** The winner of the VOT2015 competition. The method adopts pre-training of a Convolutional Neural Network (CNN) with multiple videos to learn sequence agnostic features that are common for all videos. This is achieved by having two types of layers in the training network; shared and domain-specific. On each training iteration, only one sequence and its specific fully-connected layer is enabled while shared-layers are kept the same. At the end of the training process, the domain-specific layers are swapped with a single fully-connected layer that is used in testing sequences.

**GOTURN –** The method relies completely on training data which are obtained from the ALOV300++ (Smeulders et al., 2014) and ImageNet (Russakovsky et al., 2015) datasets. During the training stage, they randomly pick pairs of successive frames and feed the ground truth bounding boxes and their augmented versions into the convolutional layers which are later used as the inputs of fully-connected layers. The method is extremely efficient running 100 FPS since it does not do need any online learning.

**TLD –** The original method adopting the three stage structure - tracking-learning-detection - that fuses the stages to correct possible errors. The tracking part tracks the given object frame-by-frame while the detector part independently searches each frame with an appearance model learned in the previous frames. Method uses two experts called *P-Expert* and *N-Expert*; *P-Expert* learns false negatives by assuming an object is supposed to follow a smooth trajectory. By utilizing this information, the tracker makes an estimation of the spatial location of the object in the next frame. If the detector part claims that there is no object in that location, it adds a positive sample into its training set. On the other hand, *N-Expert* learns false positives by using the fact that the object can be at only one location. Using outputs of the detector, the detectors choose the ones which do not overlap with the most confident result and add these patches to a training set.

**STRUCK –** A more recent implementation of the tracking-by-detection principle by adopting Structured Support Vector Machines (SSVMs) to learn how to predict object's position in the next frame instead of making a binary prediction at each candidate location. The method exploits three different types of low-level features to achieve better accuracy. First, pixel features are extracted by downsampling an image patch to $16 \times 16$ followed by a normalization

Table 2: Properties of the selected trackers.

| | C-COT (Danelljan et al., 2016) | MDNet (Nam and Han, ) | GOTURN (Held et al., 2016) | STRUCK (Hare et al., 2016) | TLD (Kalal et al., 2011) |
|---|---|---|---|---|---|
| Features | CNN | CNN | CNN | Raw Intensity, Haar, Intensity Histogram | Raw Intensity |
| Training Data | ✓ | ✓ | ✓ | | |
| Real-Time (R-T) | | | ✓ | ✓ | |
| Super R-T | | | ✓ | | |
| Implementation | Matlab | Matlab | C++ | C++ | C++ |

of the greyscale values into $[0,1]$ (256-dimensional feature). Secondly, six different Haar features are exracted to form a 192-dimensional feature vector (normalized to $[-1,1]$). The third feature set is a 16-bin intensity histogram obtained from four levels of a spatial pyramid and yielding to a 480-dimensional feature vector.

## 4.2 Performance Measures and Settings

According to the common practice in the tracking community, we adopted the weakly-correlated metrics: *accuracy* $\rho_A$ and *robustness* $\rho_R$ from (Kristan et al., 2016b). Tracker accuracy measures performance through its ability to cover ground truth bounding boxes over time and it is defined as *intersection-over-union* per frame $t$ basis

$$\phi_t = \frac{{A_t}^G \cap {A_t}^T}{{A_t}^G \cup {A_t}^T} \quad (1)$$

where $A_t^T$ is the tracker $T$ provided bounding box and $A_t^G$ the ground truth. Robustness value is the number of failures in a given sequence.

Each tracker is run five times ($i = 1,\ldots,N_{rep}$ where $N_{rep} = 5$) and during the analysis, a *burn-in region* of 10 frames as suggested in (Kristan et al., 2016b) is used. This means that the tracking results of 10 frames after re-initialization are not included in the calculations since these are tend to be biased until a certain number of frames pass. An overall result for the $i$:th run is calculated as

$$\rho_A = \frac{1}{|N_{valid}|} \sum_{t \in N_{valid}} \phi_t(i) \quad (2)$$

where $N_{valid}$ is the number of frames in which at least 50% of the object area is visible and the frame is outside of burn-in region. The average of multiple runs is equal to an average accuracy over all valid frames and repetations. Average robustness is calculated as the average of the number of failures $F$ over $N_{rep}$ independent runs

$$\rho_R = \frac{1}{N_{\text{rep}}} \sum_{k=1}^{N_{\text{rep}}} F(k) \quad (3)$$

Trackers are initialized with a ground truth bounding box in the first frame of each sequence and after each frame compared against the manually annotated ground truth. Whenever a tracking failure occurs ($\phi_t = 0$), the tracker is re-initialized on the fifth frame after the failure as recommended in (Kristan et al., 2016b) to avoid counting the same source of failure multiple times.

All trackers were executed using the default parameters and run on a Linux machine with Core i7-6700K CPU, NVIDIA GeForce GTX980Ti GPU and 16GB RAM. For computational reasons we downsampled each panorama frame to $W \times H = 2048 \times 1024$ and increased the Party video brightness by 100%.

## 4.3 Results

The evaluation results for all annotated objects in four different videos are shown in Table 3 which also includes the computation times. Weighted average per video is calculated by using the number of valid frames per video as the weigthing factor. As expected the baseline method TLD is clearly the weakest, but the more recent implementation of the same computing principle, STRUCK, performs very well as compared to the recent methods using CNN features and its C++ implementation is almost real-time. Interestingly, the VOT2015 winner MDNet outperforms the VOT2016 top performer by a clear margin. This result can be explained by MDNet's good performance on tracking people in our dataset. It is our belief that due to many slowly moving objects in their training set, their features perform better on people. GOTURN can provide fast computation (6× real-time) with the price of weaker accuracy, but still rather low failure rate (2nd best). Selection between MDNet/C-COT and GOTURN can be made based on computational limits.

## 4.4 Analysis of Tracking Failures

We analysed the root causes of the tracking failures with our dataset. For analysis we used the following methodology: for each tracking failure of the top three performing trackers (C-COT, MDNet and STRUCK), we created a mini sequence which starts 90 frames before the actual failure frame and observed the tracking results visually to understand underlying reasons of the failures. Although 90 frames might strike as too many, we believe that it is necessary since failure of a tracker is a process which starts much earlier than the actual failure. We tried

Table 3: Results for tracking objects in equirectangular panorama.

| | *Accuracy* | | | | | *Failures* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | C-COT | MDNet | GOTURN | STRUCK | TLD | CCOT | MDNet | GOTURN | STRUCK | TLD |
| Party::People | | | | | | | | | | |
| 1 | **0.689** | 0.631 | 0.436 | 0.608 | 0.616 | 2.00 | **1.00** | 2.00 | 2.60 | 15.80 |
| 2 | 0.605 | **0.691** | 0.383 | 0.647 | 0.375 | 3.00 | **0.00** | 2.00 | 2.00 | 17.40 |
| 3 | 0.344 | 0.415 | 0.329 | 0.329 | **0.459** | 1.00 | **0.00** | 1.00 | **0.00** | 14.20 |
| 4 | 0.520 | **0.640** | 0.465 | 0.410 | 0.376 | 4.00 | **0.40** | 1.00 | 1.00 | 17.60 |
| 5 | **0.687** | 0.667 | 0.471 | 0.573 | 0.577 | **1.00** | 1.00 | 2.00 | **1.00** | 27.00 |
| 6 | 0.586 | **0.636** | 0.615 | 0.472 | 0.538 | 1.00 | **0.00** | **0.00** | 4.60 | 12.80 |
| 7 | 0.584 | **0.644** | 0.407 | 0.546 | 0.499 | 4.00 | **0.00** | **0.00** | 6.40 | 29.80 |
| 8 | 0.434 | **0.691** | 0.609 | 0.498 | 0.509 | 1.00 | **0.00** | 2.00 | 2.80 | 14.40 |
| 9 | 0.570 | **0.598** | 0.472 | 0.445 | 0.529 | 2.00 | 0.00 | **0.00** | 0.20 | 16.00 |
| 10 | 0.510 | **0.527** | 0.357 | 0.294 | 0.350 | **0.00** | **0.00** | 1.00 | **0.00** | 9.20 |
| 11 | 0.795 | **0.799** | 0.237 | 0.774 | 0.481 | 3.00 | **0.00** | 2.00 | 4.40 | 13.60 |
| 12 | 0.599 | 0.619 | 0.418 | **0.637** | 0.411 | 2.00 | **0.00** | 5.00 | **0.00** | 23.00 |
| 13 | 0.610 | **0.708** | 0.500 | 0.541 | 0.420 | 3.00 | 1.40 | **0.00** | **0.00** | 14.00 |
| *w.-avg.* | 0.580 | **0.627** | 0.417 | 0.532 | 0.474 | 2.07 | **0.36** | 1.38 | 1.92 | 17.29 |
| Party::Food Cart | | | | | | | | | | |
| 1 | 0.822 | 0.786 | 0.493 | **0.841** | 0.506 | **1.00** | 1.80 | **1.00** | **1.00** | 15.40 |
| Train Station:: 9 vehicles and 2 people | | | | | | | | | | |
| Car-1 | **0.599** | 0.526 | 0.482 | 0.466 | 0.441 | 2.00 | **0.00** | 1.00 | 2.00 | 14.00 |
| Car-2 | 0.342 | 0.496 | **0.596** | 0.278 | 0.450 | 1.00 | **0.00** | **0.00** | 1.00 | 16.00 |
| Car-3 | 0.330 | **0.554** | 0.507 | 0.421 | 0.472 | **0.00** | **0.00** | 1.00 | 0.00 | 10.00 |
| Bus-1 | **0.541** | 0.531 | 0.308 | 0.344 | 0.437 | 2.00 | 2.00 | **1.00** | **1.00** | 8.00 |
| Bus-2 | 0.467 | 0.283 | **0.533** | 0.093 | 0.145 | 2.00 | 2.00 | 2.00 | **1.00** | 6.00 |
| Bus-3 | **0.697** | 0.656 | 0.382 | 0.454 | 0.498 | 2.00 | **1.00** | 3.00 | 2.00 | 12.00 |
| Minivan-1 | 0.352 | **0.443** | 0.441 | 0.294 | 0.393 | 4.00 | **2.00** | 3.00 | 3.00 | 12.00 |
| Minivan-2 | **0.672** | 0.631 | 0.621 | 0.547 | 0.483 | **0.00** | 1.00 | 1.00 | 1.00 | 4.00 |
| Minivan-3 | 0.251 | **0.484** | 0.503 | 0.286 | 0.464 | **0.00** | **0.00** | **0.00** | **0.00** | 4.00 |
| Person-1 | **0.735** | 0.652 | 0.316 | 0.467 | 0.345 | **1.00** | 2.00 | 9.00 | 3.00 | 26.00 |
| Person-2 | **0.750** | 0.689 | 0.551 | 0.548 | 0.401 | **0.00** | **0.00** | 2.00 | 2.00 | 46.00 |
| *w.-avg.* | **0.566** | 0.530 | 0.454 | 0.363 | 0.383 | 1.27 | **0.90** | 2.09 | 1.58 | 14.94 |
| Lecture::Lecturer | | | | | | | | | | |
| | 0.503 | **0.764** | 0.502 | 0.462 | 0.608 | **0.00** | **0.00** | 1.00 | 3.00 | 24.00 |
| **overall w.-avg.** | 0.607 | **0.653** | 0.440 | 0.555 | 0.505 | 1.63 | **0.60** | 1.37 | 1.84 | 17.35 |
| | Average Computation Times | | | | | | | | | |
| | Seconds per frame | | | | | × real-time (30 FPS = 1.0) | | | | |
| | 1.877 | 0.710 | 0.005 | 0.034 | 0.132 | 0.0178 | 0.0469 | 6.666 | 0.980 | 0.252 |

to identify the main failure classes, but this can still be ambiguous since multiple causes can take place simultaneously. However, we were able to define the following four main classes: *Noisy Initialization* covers the cases where initialization of a tracker is imperfect, e.g., byslight occlusion; *Occlusion* means cases where partial occlusion during tracking causes drifting and finally a failure; *Appearance Change* covers severe changes in object's appearance which might have been due to rapid illumination change, object's self-occlusion (e.g. waving hands in front of a person), or, in particular, a large view point change which often occur in spherical video; The final category is *Other* which covers the rest. Examples from each category can be seen in Figure 3 and the distributions of the failure cases for the top three trackers are given in Figure 4.

It can be observed in Figure 4 that appearance changes present the biggest tracking challenge for all top three trackers with our dataset. This can be explained by the fact that equirectangular panorama covers wide angle where appearance of an object can greatly change depending on the location of the object in the panorama. Reasons can be illumination changes in different parts of the panoramic scene, large scale changes due to equirectangular projection and, in particular, scale changes combined with 3D view point changes (see the passing bus example in Figure 3).

The bus example further confirms that the C-COT tracker is trying to find similar appearances seen earlier, e.g., the windscreen of the bus. Instead of trying to increase the scale and learn a new appearance mode, it clings on the most similar appearance to the previously seen appearances as a result of considering the tracked object as an image patch rather than an object. This is understandable as in the videos capture

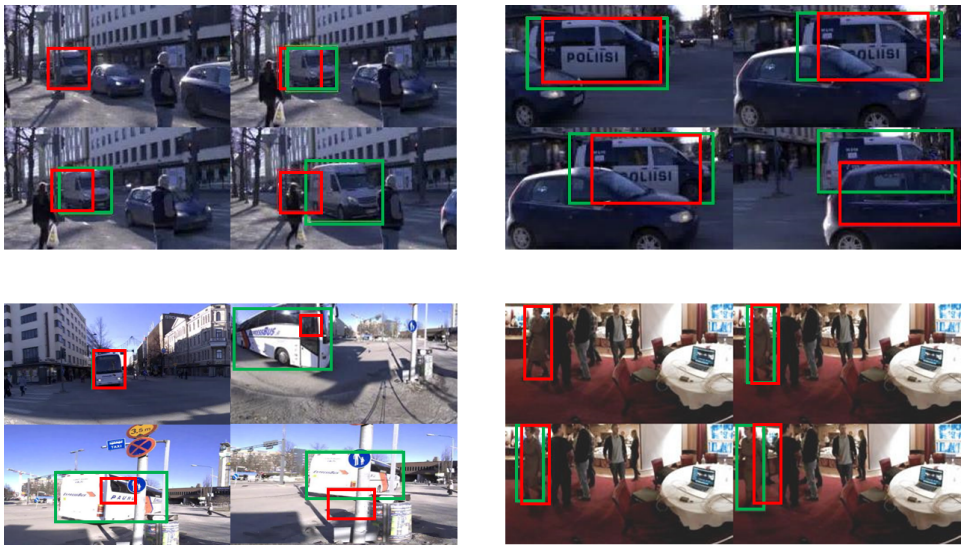


Figure 3: C-COT failures: (top left) the tracker bounding box was initialised to a region where the vehicle is occluded by trees/posts resulting to a wrong model learned (*noisy initiliasition*); (top right) the police car is occluded by another vehicle (*occlusion*); (bottom left) the bus is seen from the front, side and back view points with severe scale changes (*appearance change*); (bottom right) *other* reason for a tracking failure.

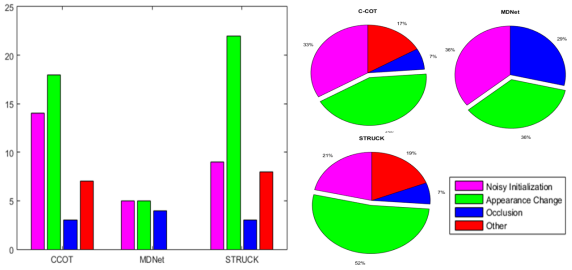


Figure 4: Distribution of Failure Causes: a. Absolute Numbers for each tracker, b. CCOT - Ratio, c. MDNet - Ratio, d. STRUCK - Ratio

with normal lenses such drastic changes rarely occur. These findings indicate that the existing trackers can be too biased on finding the most similar appearance patch in the next frame and cannot cope with rapid view point changes due to 360° spherical projection. The online learning stage could be improved by incorporating information about the current spatial location to the learning process.

## 5 Conclusion

We introduced a new, publicly available, high quality 360° dataset for benchmarking visual object tracking and detection methods. Our dataset consists of four videos with densely annotated bounding box coordinates for multiple objects in each video. We provide both the original fisheye videos from eight OZO base cameras and their geometric calibration data, and stitched equirectangular (spherical) panorama videos. To demonstrate the use of our data, we experimentally evaluated several state-of-the-art methods for visual object tracking and analysed the causes of tracking failures. Despite CCOT was state-of-the-art, our experiments showed that MDNet outperformed it with big margins in especially human based categories. As the most interesting experimental finding, large apperance changes (scale and 3D view point) are the most dominant source for tracking failures. These appearance changes are predominant for equirectangular panorama projection where appearance varies for even a same view point depending on the 3D spatial location of an object and a view point can drastically change even for straight line trajectories. We believe that better trackers for 360° video can be implemented by adapting them for these special characteristics of 360° video and in our future work we will study re-projections and view point based model updates to improve the existing trackers.

## 6 Acknowledgements

We would like to thank Prof. Atanas Gotchev, Centre for Immersive Visual Technologies(CIVIT) and Nokia Technologies for providing us with the necessary equipment assistance to capture our dataset.

# REFERENCES


Carroll, R., Agrawala, M., and Agarwala, A. (2009). Optimizing Content-Preserving Projections for Wide-Angle Images. In *SIGGRAPH*.

Cehovin, L., Lukezic, A., Leonardis, A., and Kristan, M. (2016). Beyond Standard Benchmarks: Parameterizing Performance Evaluation in Visual Object Tracking.

Corbillon, X., Devlic, A., Simon, G., and Chakareski, J. (2017). Viewport-Adaptive Navigable 360-degree Video Delivery. In *SIGGRAPH*.

Danelljan, M., Robinson, A., Shahbaz Khan, F., and Felsberg, M. (2016). Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. *ECCV 2016*.

Ferryman, J. and Ellis, A. (2010). PETS2010: Dataset and Challenge. *Proceedings - IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010*, pages 143–148.

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M. M., Hicks, S. L., and Torr, P. H. S. (2016). STRUCK: Structured Output Tracking with Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2096–2109.

Held, D., Thrun, S., and Savarese, S. (2016). Learning to Track at 100 FPS with Deep Regression Networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*.

Henriques, J., Caseiro, R., Martins, P., and Batista, J. (2014). High-Speed Tracking with Kernelized Correlation Filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):1–14.

Kalal, Z., Mikolajczyk, K., and Matas, J. (2011). Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422.

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin, L., and Vojir, T. e. a. (2016a). The Visual Object Tracking VOT2016 Challenge Results. *ECCV 2016 Workshops*, pages 777–823.

Kristan, M., Matas, J., Leonardis, A., Felsberg, M., and Cehovin, L. e. a. (2015). The Visual Object Tracking VOT2015 Challenge Results. *ICCV Workshop on Visual Object Tracking Challenge*.

Kristan, M., Matas, J., Nebehay, G., Porikli, F., and Cehovin, L. (2016b). A Novel Performance Evaluation Methodology for Single-Target Trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155.

Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Cehovin, L., Nebehay, G., and Vojir, T. e. a. (2014). The Visual Object Tracking VOT2014 Challenge Results. *ECCV Visual Object Tracking Challenge Workshop*.

Nam, H. and Han, B. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M. (2014). Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468.

Su, Y.-C. and Grauman, K. (2017). Making 360 Video Watchable in 2D: Learning Videography for Click Free Viewing. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*.

Su, Y.-C., Jayaraman, D., and Grauman, K. (2016). Pano2Vid: Automatic Cinematography for Watching 360 Videos. In *ACCV*.

Vedaldi, A. and Lenc, K. (2015). MatConvNet – Convolutional Neural Networks for MATLAB. In *Proceeding of the ACM Int. Conf. on Multimedia*.

Wu, Y., Lim, J., and Yang, M. H. Online Object Tracking: A Benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*.

Xiao, J., Ehinger, K. A., Oliva, A., and Torralba, A. Recognizing scene viewpoint using panoramic place representation. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*.