REPRODUCIBLE EVALUATION OF PAN-TILT-ZOOM TRACKING

Gengjie Chen*, Pierre-Luc St-Charles† Wassim Bouachir†
Guillaume-Alexandre Bilodeau† Robert Bergevin†
* Sun Yat-sen University, † Polytechnique Montréal, ‡ Université Laval

ABSTRACT
Tracking with a Pan-Tilt-Zoom (PTZ) camera has been a research topic in computer vision for many years. However, it is difficult to assess the progress that has been made because there is no standard evaluation methodology. The difficulty in evaluating PTZ tracking algorithms arises from their dynamic nature. In contrast to other forms of tracking, PTZ tracking involves both locating the target in the image and controlling the motors of the camera to aim it so that the target stays in its field of view. This type of tracking can only be performed online. In this paper, we propose a new evaluation framework based on a virtual PTZ camera. With this framework, we are able to replicate the main principles of online PTZ camera control and behavior including camera positioning delays, tracker processing delays, and numerical zoom. We tested our evaluation framework with the Camshift tracker to show its viability and to establish baseline results.

Index Terms— PTZ tracking, Evaluation framework

1. INTRODUCTION
Tracking with a single Pan-Tilt-Zoom (PTZ) camera has been a research topic in computer vision for many years [1, 2, 3, 4, 5]. However, it is very difficult to assess the progress that has been made because there is no standard evaluation methodology. The difficulty in evaluating PTZ tracking arises from its dynamic nature. In contrast to other forms of tracking, PTZ tracking involves both locating the target in the image and controlling the motors of the camera to aim it so that the target stays in its field of view (FOV). This type of tracking can only be performed online. As a result, it is very difficult to compare two algorithms with a real PTZ camera because the same experiment is not repeatable. Even under a strict scenario with actors performing predefined actions, the tracking conditions will never be totally identical.

Recent datasets like [6] only test the quality of the target location in each frame. Although important, it does not account for the online constraint of tracking with a PTZ camera. For example, in an online setting with a PTZ camera, if an algorithm processes a frame in one second, it is essentially blind during this entire time lapse. It means that the target may move over a large distance between two observations. Moreover, centering the camera on its previous location may result in the target leaving the FOV. In general, in PTZ tracking, algorithms follow one of two concept families:

1. Rely on a fast tracker that can process every frame without dropping any, and that always recenter the camera at the target’s previous position (which is a good approximation of its next position since the frame processing rate is high). This approach is however more likely to localize the target poorly.

2. Rely on a slower, but more sophisticated tracker that can localize the target accurately. Since being slow also means being blind for long periods of time, in order to improve robustness to fast target motion, another algorithm needs to be designed to control the camera. A typical approach is to determine the target’s most probable location in the next frame, and center the FOV on that position.

In short, the processing time budget is important in PTZ tracking because of its online nature, and slow processing means missed observations, which might be crucial for accurate results. With this paper, we hope to inspire the development of better PTZ tracking methods by proposing a virtual camera that allows panning, tilting, and zooming inside pre-recorded spherical panoramic videos. Under our new proposed evaluation framework, tracking conditions do not change for each experiment. We replicate the main principles of online PTZ camera control behavior by considering camera positioning delays, tracker processing delays, and numerical zoom. In this work, we focus on single object tracking, but our framework can also be used for multiple object tracking. Our contributions are: 1) a publicly available C++ library implementing a virtual PTZ camera. It offers basic functionalities (image acquisition, camera movement) as well as online evaluation of tracking performance using four metrics; 2) three publicly available spherical panoramic scenarios taken in two real-world environments, featuring a total of 36 manually annotated tracking sequences for various object types; and 3) a set of baseline performance results obtained using the Camshift tracker [7]. Note that an extended version of this paper is available online.

2. RELATED WORK
To the best of our knowledge, only two works specifically addressed the evaluation of tracking with a PTZ camera [8, 9]. In the work of Qureshi and Terzopoulos [9], a virtual world is simulated where animated pedestrians can be tracked by various virtual sensors, including virtual PTZ cameras. This approach is very interesting as it allows repeatable evaluation. Its drawback is that it does not reproduce real-world settings such as change in lighting conditions, nor addresses the limits of real camera sensors (resolution, motion blur, etc.) because the scenes are artificial. It was used in the context of sensor networks. Salvagnini et al. [8] proposed an experimental framework where a real PTZ camera tracks objects moving on a large screen. The goal of this work was the same as ours: it does provide repeatable scenarios for internal use in a given research laboratory, but other research groups cannot repeat the same experiments as they are equipment-specific. Furthermore, the PTZ camera motion is limited to a very small portion of its operating sphere.

The evaluation metrics used in previous work on tracking with a PTZ camera are varied but are essentially very similar to those for the

*This work was conducted while Gengjie Chen was doing a MITACS Globalink internship at Polytechnique Montréal

1http://www.polymtl.ca/litiv/vid/index.php
2http://arxiv.org/abs/1505.04502
evaluation of single object trackers or multiple object trackers. For example, in Cai et al. [5], tracking is evaluated with multiple object tracking metrics. There is no specific evaluation of camera control, although for this work the PTZ camera mostly zooms (it does not pan or tilt significantly). However, most authors evaluate camera control to some extent. In both the work of Lee et al. [10] and Liu et al. [11] tracking is evaluated by the percentage of frames where the tracked object is in the FOV. Such a metric roughly evaluates both camera control and tracking performance simultaneously.

Since tracking with a single PTZ camera requires the evaluation of both tracking and camera control performance, Darvish and Bilodeau [4] and Salvagnini et al. [8] proposed metrics for both aspects. Center Location Error (CLE) and overlap ratio [12] were used for tracking accuracy, and the distance between the center of the ground-truth target position and the center of the image was used to evaluate camera control. The assumption for the evaluation of the camera control is that if it is done properly, the target will always be close to the center of the FOV. If not, the probability that the target will leave the FOV after sudden movements or direction changes is high. In addition to those metrics, Darvish and Bilodeau [4] and Paillet et al. [13] also included a track fragmentation metric, that is, the number of frames for which the target is out of the FOV.

3. EVALUATION FRAMEWORK

Our PTZ evaluation framework has three components: 1) a C++ library that simulates the main behavior of a PTZ camera and includes an evaluator, 2) a collection of spherical panoramic videos for different scenarios, and 3) their corresponding ground-truth (GT) annotated sequences. The PTZ simulator grabs panoramic images from a video file, builds the scenario model and provides a typical viewing frustum for the tracker based on camera parameters. The evaluator uses basic ground-truth data and the same camera parameters to generate ground-truth bounding boxes for the current FOV, and then compares them with actual tracking results.

The framework has been designed based on videos captured by a Point Grey Ladybug 3 Spherical camera and OpenGL to project the videos on a sphere. The Ladybug camera gives a near 360° spherical view of the scene that can be mapped on such a surface. It is thus possible to design a virtual camera that can observe specific portions of the sphere. Therefore, we obtain a virtual PTZ camera that can be controlled as desired to track objects in pre-recorded videos. For convenience, the center of the spherical model is set at the origin of the world coordinates.

3.1. PTZ Camera Model

After building the model of the scenario, the virtual camera is placed at the origin $O$, as shown in Fig. 1. Although the position of the camera is constrained, it still has three degrees of freedom. They are: 1) pitching, 2) yawing, and 3) rolling, or the rotation on the axis between $O$ and the targeted point, $D$. Since we are simulating a PTZ camera, rolling is ignored and changing the pitch and yaw angles achieves the functionality of tilting and panning, respectively. Consequently, we use the normal vector $OD$ of the image plane to define the direction vector of the camera in world coordinates. This direction is determined by the tilt angle $\theta_d$ and the pan angle $\phi_d$ (Fig. 1).

A perspective projection model is used to transform points on the image plane to points in world coordinates and vice-versa. This allows us to calculate the part of the view sphere that should be projected on the image plane of the virtual camera. The virtual PTZ camera returns images based on its FOV ($\alpha$) and orientation ($\theta_d, \phi_d$) parameters. It can provide images only when it is still. We preferred this option to artificially generating motion blur. Although debatable, previous works seem to agree on the fact that images with strong motion blur are not really usable [4].

In our implementation, the virtual PTZ camera can have its orientation changed either by using specific pan $\theta_d$ and tilt $\phi_d$ angles, or by recentering on a pixel position expressed in image coordinates. The zoom can also be simulated by changing the vertical FOV angle of the camera ($\alpha$). To reproduce the behavior of an actual PTZ camera, we consider that orientation changes are not instantaneous. Instead, the simulated camera pans and tilts based on the maximal angular speeds of a commercial PTZ camera (Sony SNC-RZ50N). This means that the image acquisition delay after a reorientation depends on the amplitude of camera’s motion. To simulate this first type of delay (noted $\tau_m$), frames are simply skipped in the video.

A second type of delay, noted $\tau_p$, corresponds to the time required for a tracker to process a frame. We also consider a third type of delay, $\tau_c$, which is the communication delay over a network in the case of an IP PTZ camera. The user fixes this last delay. Note that all delays are simulated by skipping frames in the pre-recorded videos to mimic dropped frames. This is justified by the fact that, in an online scenario, frames cannot be delayed in a buffer for later processing.

Following a camera motion, the tracker will observe the scene again after a $\tau = \tau_m + \tau_p + \tau_c$ delay. Therefore, ideally, a tracker should try to minimize $\tau_p$ as much as possible and also try to predict the position of the target after the $\tau$ delay to make sure it is within its FOV.

3.2. Performance Evaluation

Apart from the basic operations described in the last section, our PTZ camera framework also calculates four performance metrics to evaluate a tracker. Let $c_{GT}^t$ and $c_{PT}^t$ be the center locations of the ground-truth target and the predicted target (by the tracker) at time $t$, respectively. $A_{GT}^t$ and $A_{PT}^t$ be the bounding boxes of the ground-truth target and the predicted target at time $t$, respectively, and $c_{FOV}^t$ be the location of the center of the image FOV at time $t$. CLE (Center Location Error) and OR (Overlap Ratio) at time $t$ evaluate the quality of target localization and are defined as

$$CLE^t = |c_{GT}^t - c_{PT}^t|$$

$$OR^t = \frac{A_{GT}^t \cap A_{PT}^t}{A_{GT}^t \cup A_{PT}^t}$$
Many perturbation factors can affect tracking performance. For our dataset, we used the difficulty categorization proposed in [6]. Three tracking difficulties are present in all our sequences: Motion Blur (MB), Scale Change (SC), and Out-of-Plane Rotation (OPR). Moreover, we defined subsets of videos corresponding to other perturbation factors: Fast Motion (FM), Cluttered Background (CB), Illumination Variation (IV), Low Resolution (LR), Occlusion (OCC), presence of Distractors (DIS), and Articulated Objects (AO). The histogram of Fig. 2 illustrates the difficulty distribution in our dataset. Note that one tracking sequence may include multiple difficulties. Fig. 3 shows examples of targets that are tracked in our dataset. 

![Fig. 2. Difficulty distribution over the whole dataset.](image)

![Fig. 3. Examples of tracked objects in the proposed dataset.](image)

### 3.3. Spherical Panoramic Scenarios

Three spherical video sequences were captured in two indoor environments with four or five randomly moving individuals. The videos contain a total number of 3,179 panoramic frames recorded at a frame rate of 16 fps (the maximum frame rate of the Ladybug 3). The first video was captured in a laboratory room cluttered with desks, chairs, posters and technical video equipment in the background. The Ladybug 3 camera was mounted on a tripod and placed in the center of the room. The other two spherical videos were recorded in the middle of a large atrium within a building with glass walls causing uneven illumination conditions. Our complete dataset includes 36 manually annotated tracking sequences extracted from the three initial spherical video sequences. The length of each tracking sequence varies from a few seconds to one or two minutes. For each of the 36 annotated sequences, the tracked target is one of the following: the full body of a moving person, a torso, a head, or an object carried by a person.

Many perturbation factors can affect tracking performance. For our dataset, we used the difficulty categorization proposed in [6]. Three tracking difficulties are present in all our sequences: Motion Blur (MB), Scale Change (SC), and Out-of-Plane Rotation (OPR). Moreover, we defined subsets of videos corresponding to other perturbation factors: Fast Motion (FM), Cluttered Background (CB), Illumination Variation (IV), Low Resolution (LR), Occlusion (OCC), presence of Distractors (DIS), and Articulated Objects (AO). The histogram of Fig. 2 illustrates the difficulty distribution in our dataset. Note that one tracking sequence may include multiple difficulties. Fig. 3 shows examples of targets that are tracked in our dataset.

\[ OR_t^i = \frac{A_{GT_t} \cap A_{PT_t}}{A_{GT_t} \cup A_{PT_t}} \]  

\[ TCE_t^i = |c_{FOV}^t - c_{GT}^t| \]  

\[ TF_t^i = \begin{cases} 1 & \text{if } CLE_t^i \text{ is invalid} \\ 0 & \text{otherwise} \end{cases} \]

TF indicates whether the target is inside or outside the FOV. CLE and TCE are invalid and assigned -1 if the target is outside the FOV. The overall metrics \( CLE, TCE, \) and \( OR \) are the averages all valid \( CLE, TCE, \) and \( OR \) metrics, respectively. \( TF \) is the sum of all \( TF \) divided by the number of processed frames.

### 3.4. Ground-Truth

The GT annotations for a PTZ camera video are very different from those of a fixed camera video. A traditional camera with a fixed view frustum simply requires a sequence of bounding boxes for each tracked object, which can be defined by three parameters: width, height, and 2D center position inside the frame. A PTZ camera GT is much more complex: setups with different FOVs and orientations as well as output image sizes may still be able to observe the same target. It is thus necessary to make the GT applicable to all possible observation configurations. To do so, we first manually annotate a "basic" GT sequence from which the actual GT annotations required for evaluation can be obtained by projection and rectification operations. In this basic GT, four values are recorded for each frame of the video: two are related to the current orientation of the camera when it is centered on the target, i.e. the pan and tilt angles. The other two are simply the width and height of the target’s bounding box. While collecting basic GT, all other camera parameters are fixed, but they must also be recorded since they are necessary for GT transformation. These parameters are the camera’s vertical FOV angle and its output image width and height. Then, in order to obtain the GT coordinates of the target on the image plane of a camera with different parameters, we use the projective projection model and do coordinate transformations based on the actual camera parameters and recorded GT parameters.

### 4. Baseline Evaluation Results

In order to provide baseline tracking results, we used our virtual camera framework to evaluate a simple PTZ tracker based on the well-known Camshift algorithm [7]. Looking back at the two general PTZ tracking concept families described in section 1, we can classify this tracker as part of the first family (i.e. fast, but not very robust). As such, we used the typical camera control strategy of this concept family, meaning that the camera FOV is continuously centered at the target’s previous location.

We tested the Camshift tracker on the proposed 36 sequences at a 640x480 resolution with a 90° vertical FOV angle, using the categorized difficulties defined in section 3.3. Tables 1, 2, 3, and 4 present the results of Center Location Error (CLE), Target to Center Error (TCE), Overlap Ratio (OR), and Track Fragmentation (TF), respectively. In all these experiments, we simulated the camera motion delay \( \tau_m \) of the commercial PTZ camera Sony SNC-RZ50N based on its maximal angular speed of 300°/s. We considered the processing delay of the tracker \( \tau_p \) as the actual execution time of Camshift for each frame. However, since the Camshift tracker can typically process more than 16 fps (we ran it on an Intel i5 3570 CPU...
In DIS, all targets are human heads, which are rather easy to track due to their varying sizes and overlaps and the nature of their targets, we note that the scores obtained for all four metrics in the Distractors (DIS) category are generally better than those of any other category. In DIS, all targets are human heads, which are rather easy to track with a histogram-based method, as long as the background does not match skin color. The Illumination Variation (IV) category seemed to be the hardest to handle for Camshift; extreme contrast and camouflage problems due to intense light sources sometimes made tracking nearly impossible. The Low Resolution (LR) and Articulated Objects (AO) categories share multiple sequences with IV, which might explain their similar scores.

More generally, we can observe that increasing the communication delay \( \tau_c \) has a deep impact on the effectiveness of the tracking algorithm. While it could be expected that increasing this parameter’s value would directly worsen tracking performances, interestingly, this is not always the case. In fact, for a handful of sequences presenting full occlusions, adding a communication delay sometimes helps the tracker by either completely eliminating these occlusions or by replacing them with partial occlusions. This is however uncommon. More typically, all metrics except Track Fragmentation \( TF \) show decreases in performance for each increment of \( \tau_c \).

### 5. Conclusion

In this paper, we have proposed a new publicly available framework for the evaluation of PTZ tracking algorithms. It allows realistic experiments to be repeated in identical conditions. This framework simulates a PTZ camera that can pan, tilt, and zoom to observe different parts of a scene constructed using pre-recorded spherical panoramic videos. It also considers various types of delays and limitations of commercial PTZ cameras to provide a reproduction of real tracking experiments. We provide a total of 36 annotated tracking sequences along with our PTZ framework, which sum up to over 16,000 bounding boxes. To provide baseline results for our framework, we tested the Camshift algorithm on these 36 sequences. The four metrics we use to evaluate PTZ tracking performance indicate that Camshift is generally unable to handle the challenges present in typical PTZ tracking scenarios. We are confident that tracking methods specifically designed for PTZ scenarios can overcome the realistic difficulties present in our dataset.
6. REFERENCES


