Robust and Enhanced 360° Visual Tracking based on Dynamic Gnomonic Projection

Hao Peng[a], Yun Zhang[b] and Fang-Lue Zhang[a,*]

[a]School of Engineering and Computer Science, Victoria University of Wellington, New Zealand; [b]College of Media Engineering, Communications University of Zhejiang, China

ARTICLE HISTORY
Compiled May 28, 2025

ABSTRACT
Recently, 360-degree visual tracking has become increasingly important in 360-degree video processing technology. Although visual tracking technology in 2D videos has gradually matured, there is no universal method for visual tracking in 360-degree videos that can effectively address image stretching and object deformation caused by the equirectangular representation of 360-degree images. In this paper, we propose a two-part method for 360-degree visual tracking. The first part is a general method that can be integrated into any 2D visual tracking system to be applied to 360-degree videos. This part converts equirectangular images into 2D gnomonic projections, enabling the use of existing 2D tracking algorithms while mitigating image distortion. Then, building upon the UPDT algorithm, the second part integrates the general 360-degree visual tracking method with additional enhancements to improve robustness and efficiency in 360-degree visual tracking. Furthermore, when tracking performance deteriorates, it combines results from the sample set and trajectory prediction to achieve more robust and accurate tracking. In our experiments, We use two datasets in 360-degree equirectangular representation to demonstrate the effectiveness and advantages of our proposed method. Additionally, we explore the application of 360-degree visual tracking methods in editing, enabling the automatic manipulation of moving objects in 360-degree videos.

KEYWORDS
360-degree; Tracking; Equirectangular; Gnomonic; Robust

1. Introduction

Visual tracking, first theorized by Wax in 1955 (Wax 1955), has become a key research area in image processing with significant practical applications. Given the target's initial state in the first frame, tracking methods estimate its position throughout the video sequence. Based on the number of tracked targets, visual tracking can be categorized into single-object and multi-object tracking. This paper primarily focuses on single-object tracking, where a single instance of an object class is monitored. [1]

A 360-degree (360°) image, also known as a spherical image, is a crucial component in 360° video processing. However, existing image processing methods are generally based on two-dimensional images. To adapt to the 360° image processing pipeline, it is necessary to convert spherical images into two-dimensional planes while preserving

[1]*Fang-Lue Zhang is the corresponding author.

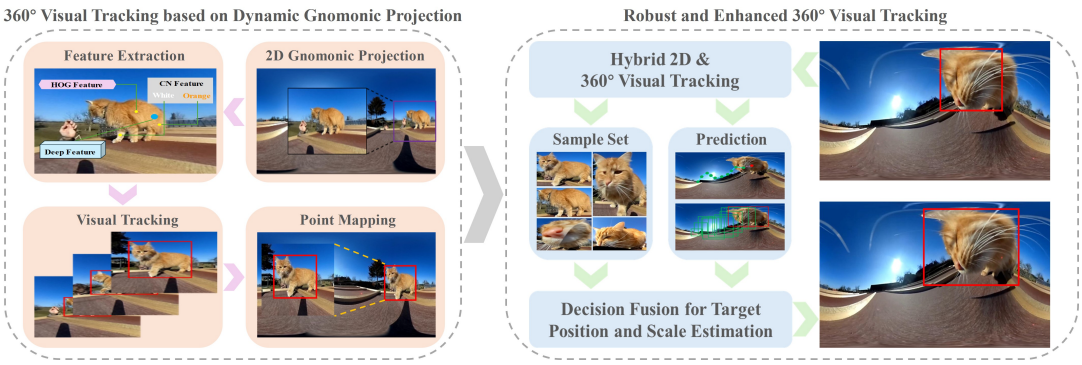

Figure 1. The overall workflow of our two-part method for 360° visual tracking. The first part (left side of the image) is a general 360° visual tracking approach based on dynamic gnomonic projection. The second part (right side of the image) builds upon the first and provides a more robust and enhanced 360° visual tracking solution.

(a) Spherical representation (b) Equirectangular representation (c) Cubemap representation

Figure 2. Different spherical image representation methods.

omnidirectional information. The three most widely used image representation methods are spherical representation (Figure 2a), equirectangular representation (Figure 2b), and cubemap representation (Figure 2c) (da Silveira et al. 2022). In these methods, the equirectangular representation can be treated as a standard 2D projection of a 360° image and processed by existing algorithms for feature extraction and matching. Therefore, we generally use the equirectangular representation in 360° visual tracking for videos.

Some advancements in 360° video capture and VR/AR display technologies have enabled increasingly immersive visual experiences. Prior research has begun to explore interaction techniques that enhance user experience in this context. For example, Li et al. investigated how bullet comments can be effectively displayed and inserted in immersive 360° video environments. (Wang et al. 2020) Their user studies showed that spherical sliding comments significantly improve user engagement and social interactivity and that intuitive drag-based insertion methods are generally preferred. Complementarily, (Li et al. 2022b) proposed Transitioning360, a system that enables efficient 360° video playback on 2D displays through content-aware NFoV (Normal Field-of-View) camera paths and spatially-aware transitions. This approach improves users' ability to locate and follow relevant content while minimizing cognitive load. Together, these works highlight the importance of designing interaction models that balance freedom of exploration with guided navigation and social presence in 360° video experiences.

In recent decades, significant progress has been made in single-object tracking with the development of numerous advanced algorithms (Song 2014; Liao et al. 2020; Zhou et al. 2022; Hong et al. 2024). These technologies are widely used in intelligent monitor-

ing, human-computer interaction, military guidance, and other fields. However, challenges such as illumination changes, deformation, and occlusion make visual tracking a complex and ongoing research area (Yang et al. 2011). Nowadays, with advancements in computing power, Virtual Reality (VR) and Mixed Reality (MR) technologies have also evolved (Friston et al. 2019; Tursun et al. 2019). VR 360° videos (VR360°), offering a full 360°×180° field-of-view (Schroers et al. 2018), have gained increasing attention due to their immersive experience and applications in entertainment, education, tourism, and healthcare. Their integration with artificial intelligence and computational photography presents new research opportunities. However, 360° visual content, stored using equirectangular projection, introduces geometric distortions that affect tracking accuracy (Coors et al. 2018; Li et al. 2022a; Wang et al. 2024). Addressing these distortions remains a key challenge, and current 360° tracking performance remains suboptimal. Additionally, existing 360° dynamic object editing methods struggle with moving objects and boundary distortions, often leading to editing failures.

In this paper, we explore a method to enhance 360° visual tracking and investigate its application in dynamic object editing for 360° videos. We first employ the gnomonic projection method to propose a general 360° visual tracking approach and integrate it into the UPDT method (Bhat et al. 2018). Then, we improve the 360° visual tracking method by integrating two approaches. The improved method first combines UPDT-based 360° tracking with conventional visual tracking, applying the original UPDT method to the central region of the image while using the UPDT360 method for boundary regions. Additionally, we refine the scale computation for target width and the Field of View (FoV), significantly enhancing robustness and computational efficiency. Building upon this, we introduce specialized enhancements tailored for 360° visual tracking. The tracking quality is assessed by analyzing peak response values and the number of secondary peaks in the response map. Moreover, a sample set strategy is incorporated to mitigate sample contamination issues in complex tracking scenarios. For frames with challenging conditions, a Kalman filter is employed to predict target position and scale. By combining sample set results with target predictions, overall tracking performance is further improved. We conduct extensive experiments and ablation studies to evaluate our method, analyzing its results and algorithmic parameters. Experimental findings demonstrate that our proposed method significantly enhances tracking accuracy and efficiency in 360° ERP videos. Furthermore, we apply our tracking approach to dynamic object editing in 360° videos, enabling direct modifications to moving objects while effectively addressing boundary and distortion issues caused by the ERP representation. Figure 1 illustrates the overall workflow of our method. The key contributions of this work are as follows:

- We propose a general 360° visual tracking method based on the gnomonic projection, enabling conventional 2D tracking algorithms to be applied to 360° ERP videos while mitigating distortions.
- We enhance 360° visual tracking by integrating a dynamic bidirectional projection approach with a trajectory-aware sample set strategy, combining 2D and 360° tracking with motion modeling and Kalman filter-based prediction to improve robustness and efficiency.
- We apply 360° visual tracking results to dynamic object editing, overcoming the limitations of existing 360° video editing techniques that cannot directly edit moving objects.

This work advances both 360° visual tracking and video editing, addressing critical

challenges and paving the way for more effective 360° video processing.

2. Related Work

For better understanding, in this section, we revisit the background of visual tracking, including Discriminative Correlation Filters (DCF)-based methods and deep learning-based methods. Additionally, we discuss existing efforts in 360° visual tracking.

2.1. Visual Tracking

Trackers based on Discriminative Correlation Filters (DCF) have always been key methods in visual tracking. Compared to traditional tracking algorithms based on object detection (Song 2014), DCF improves computational efficiency and robustness by solving the ridge regression problem using circular structures in the frequency domain. Early DCF methods like MOSSE (Bolme et al. 2010), KCF(Henriques et al. 2014), and Staple (Bertinetto et al. 2016) have all enhanced the reliability of visual tracking while achieving online tracking. Additionally, SAMF (Li and Zhu 2015) and DSST (Danelljan et al. 2014) have incorporated scale processing. BACF (Schroers et al. 2018) improves the quality of extracted features by using HOG features. In DCF methods, the two main issues affecting visual tracking are the boundary effect and temporal filter degradation. Many methods have successfully utilized guidance to address these two issues and serve as prior models for visual tracking. To solve the first issue, the boundary effect, the Spatially Regularized DCF (SRDCF) (Danelljan et al. 2015) introduces penalties for the background when training correlation filters. Building on this, the Spatio-Temporal Regularized DCF (STRCF) in (Li et al. 2018) introduces spatio-temporal regularization to obtain a joint solution for the two main problems, achieving better performance than SRDCF. (Zhu et al. 2021) proposes a bilateral weighted regression sorting model with spatio-temporal correlation filters, further improving tracking accuracy. (Danelljan et al. 2016) introduces sub-pixel tracking through learning Continuous Convolution Operators (CCOT). Efficient Convolution Operators (ECOs) (Danelljan et al. 2017) are proposed to achieve a lightweight version of CCOT with generative sample space and dimensionality reduction mechanisms.

Moreover, with the continuous development and wide application of deep learning theories in recent years, some researchers have also begun to integrate deep learning into visual tracking algorithms (Wang et al. 2018; Hu et al. 2018). Currently, the application of deep learning in RGB visual tracking can be roughly divided into two types: one is to apply deep learning to feature extraction and use correlation filtering as the framework for visual tracking methods; the other is purely based on neural network frameworks for visual tracking. Representative methods of the latter include DiMP (Bhat et al. 2019) and TransT (Chen et al. 2021). DiMP improves tracking performance by learning a discriminative target model through online optimization. By leveraging a transformer-based architecture, TransT effectively fuses target and template features for robust tracking.

By combining deep features and handcrafted features, the UPDT (Bhat et al. 2018) algorithm improves tracking accuracy and robustness through the reasonable application of combined features.

2.2. 360° Visual Tracking

Currently, there are limited efforts dedicated to integrating planar object editing and tracking into 360° videos. In the field of 360° visual tracking, researchers have adapted methods originally designed for 2D videos to address the unique challenges posed by 360° data. For example, (Cai et al. 2018) combines multi-scale kernelized correlation filters (KCF (Henriques et al. 2014)) with Kalman estimation to enhance scale handling and occlusion detection, using the peak sidelobe ratio (PSR) for identifying occlusions and resuming tracking once occlusion ends. Similarly, (Delforouzi et al. 2016) focuses on improving tracking performance for unknown objects in 360° camera images, tackling challenges like non-planar rotations and complex backgrounds by refining detectors and classifiers. Another approach, described in (Delforouzi et al. 2020), integrates Kalman filters and the Lucas-Kanade method to address tracking challenges specific to 360° videos, leveraging YOLO and deep learning-based object detectors to extract object priors and enhance tracking robustness. Meanwhile, (Mi and Yang 2019) evaluates the performance of eight state-of-the-art tracking algorithms on 360° videos, identifying key challenges such as viewpoint changes, occlusions, deformations, lighting variations, scale changes, and camera shake.

Regarding the dataset, the 360VOT dataset (Huang et al. 2023) serves as a comprehensive benchmark for omnidirectional visual tracking. It consists of 120 high-resolution video sequences covering diverse scenarios and tracking targets across 32 categories. Additionally, it provides four types of ground truth annotations, introducing new evaluation metrics for 360° visual tracking.

3. Proposed Method

In this section, we propose a two-part visual tracking method for 360° ERP videos. It is a 360° visual tracking method based on dynamic gnomonic projection. The first part utilizes 2D gnomonic projection and point mapping to extend any 2D visual tracking approach for 360° visual tracking. The second part builds upon and enhances the first part, refining the UPDT-based 360° tracking framework to improve robustness, efficiency, and performance under complex tracking conditions.

3.1. 360° Visual Tracking with Dynamic Gnomonic Projection

Our approach in the first part presents a 360° visual tracking method using dynamic gnomonic projection, designed to address the distortions and spatial complexities introduced by equirectangular projection (ERP) images. By integrating spherical image transformation with conventional 2D visual tracking techniques, this method ensures accurate and seamless tracking in 360° video environments. Given its effectiveness and ease of implementation, it serves as the foundation for our subsequent work. This method is also referenced in our previously submitted conference paper (Peng and Zhang 2024).

3.1.1. Overview

The pipeline of the first part of this method is shown in the left half of Figure 1. Starting from the first frame, the initial target position is obtained in the original equirectangular projection (ERP) image. This serves as the starting point for subse-

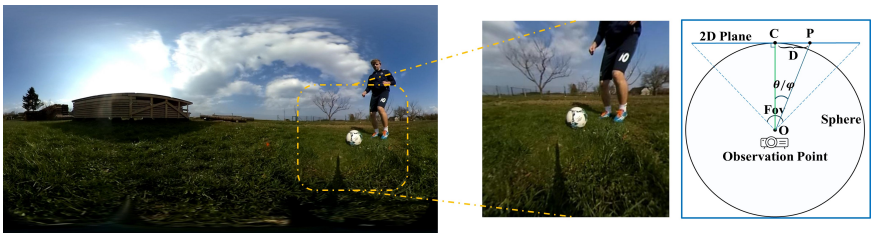

Figure 3. 360° ERP image transform to 2D gnomonic image.

quent operations. The identified target position is used as the focal point to project the ERP image into a 2D gnomonic image with a fixed 90° Field-of-View (FoV), ensuring a locally planar view that mitigates distortions inherent in the spherical representation. The corresponding center and top-left corner of the target's position in the ERP image are mapped onto the 2D gnomonic image for accurate localization.

Next, this method maps the corresponding points of the center and the top-left corner of the target position in the ERP image onto the 2D image. The tracking features of the target object are then extracted from the 2D image, which includes HOG features, CN features, and deep features to enhance tracking robustness. The 2D visual tracking algorithm is then applied to the gnomonic image, utilizing the extracted features to determine the new target position in the transformed 2D space.

For subsequent frames, the previously tracked target position serves as the reference point for transforming the next frame's ERP image into a 2D gnomonic projection with a fixed FoV. The 2D tracking method is continuously applied to the newly generated 2D image, yielding an updated target position. The center and top-left corner of the tracked target are then mapped back to the ERP image to determine the new bounding box in the original 360° space. This process is repeated iteratively for each frame until the video concludes.

By leveraging 2D gnomonic projection, this method effectively reduces distortion and enhances the applicability of conventional 2D tracking algorithms in the spherical domain of 360° videos. Additionally, mapping the tracking results back to the ERP representation ensures consistency and seamless integration within the original 360° image format. The combination of projection, feature extraction, 2D visual tracking, and bidirectional point mapping forms the core of our approach, ensuring robust and precise visual tracking across the entire 360° video.

3.1.2. Local 2D Gnomonic Projection

As illustrated in Figure 3, the first step in achieving 360° ERP visual tracking is to generate a 2D gnomonic projection from each frame of the original ERP image. To implement a Local 2D Gnomonic Projection, we need to determine the Field of View (FoV) and the projection center. This image represents a 2D embedded plane with a fixed FoV angle, where the observer's viewpoint is positioned at the sphere's center (O Point). Following the approach outlined in (Guo et al. 2022; Regensky et al. 2022), this method transforms the ERP image into a 512×512 2D representation, where the user's observation point is positioned at the center of the sphere.

The initial FoV is set to 90°, considering that target position changes between consecutive frames are generally small. Even for fast-moving targets, it is unlikely that they will move out of the image range within a single frame when using a 90° FoV. Additionally, due to the nature of gnomonic projections, a larger FoV would introduce significant distortion. Therefore, we set the initial FoV to 90° and dynamically adjust

the viewport's FoV to 60° or 120° when the target's length or width falls below or exceeds predefined thresholds. The 2D gnomonic images have a resolution of 512×512 pixels, with thresholds set at 20 and 400 pixels, respectively, based on experimental results. For the projection center, we select the center coordinates of the tracked object from the previous frame to ensure that in the next frame, the target remains near the center of the 2D image while maintaining a sufficient FoV.

3.1.3. Mappings between the ERP and the Gnomonic Images

In the first frame, after generating the 2D gnomonic image centered on the target, the center and top-left corner of the target in the ERP image need to be mapped to the 2D image. In the 2D image, the target's width is twice the difference between the top-left corner's x-coordinate and the center's x-coordinate, while the height is twice the difference between the top-left corner's y-coordinate and the center's y-coordinate. After tracking is completed in the 2D domain, the results are mapped back to the ERP image.

First, the center of the 2D gnomonic image is set as the center of the ERP image, and calculations are performed based on the tracking results from the previous frame. Since the center of the 2D image corresponds to the target center in the ERP image, this method follows the same approach used for generating the gnomonic image to project the target's top-left corner onto the 2D gnomonic image. As illustrated in the geometric diagram in Figure 3, the length of OC can be calculated based on the FoV angle as:

$$OC = \frac{a}{2\tan\left(\frac{Fov}{2}\right)} \tag{1}$$

For the n-th frame, let the point of interest in the 2D image be (x_n^{p2D}, y_n^{p2D}), and the 2D image center be (x_n^{c2D}, y_n^{c2D}). Then, the method computes the target point's horizontal angle θ and vertical angle ϕ relative to the image center using trigonometric functions:

$$\theta = \arctan\left(\frac{x_n^{p2D} - x_n^{c2D}}{OC}\right), \quad \phi = \arctan\left(\frac{y_n^{p2D} - y_n^{c2D}}{OC}\right) \tag{2}$$

Finally, the corresponding positions of the target center and top-left corner in the ERP image are obtained by adjusting the previous frame's target center with the angle changes θ and ϕ, ensuring accurate mapping between the 2D gnomonic image and the ERP image, thereby improving tracking accuracy in 360° videos.

3.1.4. Scale Calculation and Boundary Case

In 360° ERP visual tracking, the width of the ERP image is not entirely independent of latitude, as it gradually contracts toward the poles. To estimate the scale of the target in the tracking frame, adjustments must be made based on latitude. Given the target's center and top-left coordinates (x_n^{cERP}, y_n^{cERP}) and $(x_n^{tlERP}, y_n^{tlERP})$, the actual target width in the ERP image is computed as:

$$W_{\text{target}}^{ERP} = 2 \cdot \text{abs}(x_n^{tlERP} - x_n^{cERP}) \cdot \cos^{-1}\left(\frac{\text{abs}\left(\frac{H^{ERP}}{2} - y_n^{cERP}\right) \cdot \pi}{H^{ERP}}\right) \tag{3}$$

Here, the height of the ERP image and the width of the target in the ERP image are denoted as H^{ERP} and W_{target}^{ERP} respectively. Additionally, special handling is required for boundary conditions. The left and right boundaries are cyclically connected, meaning objects exiting one side reappear on the opposite side. The top and bottom boundaries follow a symmetry rule, where objects moving past them reappear at a position mirrored across the centerline. These adjustments ensure accurate tracking within the ERP image framework.

3.2. Robust and Enhanced 360° Visual Tracking

While recent deep learning-based trackers such as DiMP (Bhat et al. 2019) and TransT (Chen et al. 2021) have demonstrated high accuracy in visual tracking, our method continues to rely on Discriminative Correlation Filter (DCF)-based frameworks. This choice was made primarily due to efficiency concerns. Deep learning-based methods typically require significant computational resources and exhibit slower inference speeds, which makes them less suitable for real-time applications, especially in the context of 360° visual tracking that already involves computationally intensive operations like spherical-to-planar projection and coordinate transformations. In contrast, DCF-based trackers offer a good balance between accuracy and speed, and their lightweight structure allows for smoother integration into our two-stage 360° tracking framework. Furthermore, our proposed enhancements—including hybrid tracking strategies and adaptive projection—focus on improving robustness and efficiency without relying on heavy neural network architectures. As a result, the DCF-based approach remains more practical for our target use case. Among DCF-based methods, we chose UPDT (Bhat et al. 2018) as our baseline due to its favorable balance between tracking accuracy and speed, achieved through the effective combination of deep and handcrafted features.

The method introduced in the last section improves upon traditional 2D tracking by addressing cross-border issues and latitude distortion in 360° ERP images. While UPDT360 achieves the best results on 360° ERP datasets, its robustness remains a concern, as it can underperform compared to 2D methods in cases of drift or positional errors. The reliance on 2D gnomonic projection restricts the search area, making target recovery difficult, while frequent updates in correlation filter-based tracking can lead to learning irrelevant content. Additionally, the method lacks strategies for handling deformation and occlusion, and the need for gnomonic projection in each frame increases computational cost, reducing efficiency by 20%-50%. To overcome these limitations, this chapter explores strategies to enhance the robustness, efficiency, and overall performance of 360° visual tracking.

In this subsection, we propose two steps to enhance the existing UPDT-based 360° visual tracking method (UPDT360). The pipeline of this part is shown in the right half of Figure 1. In the first step, by combining spherical image transformations with conventional 2D visual tracking techniques, UPDT360 is optimized to improve robustness and efficiency. However, several challenges in 360° video visual tracking remain unresolved, including deformation, occlusion, motion blur caused by fast movements, and

difficulty in tracking small targets. These issues occur more frequently in 360° visual tracking, which can result in tracking failures or drifts, significantly limiting overall performance. To address these problems, in the second step, we integrate the improved method with sample sets and target prediction, enhancing its ability to maintain stable performance under complex tracking conditions.

3.2.1. Hybrid 2D and 360° Visual Tracking Method

To strengthen robustness, it is necessary to expand the search area, ensuring that tracking is not confined solely to the 2D projection image in certain frames and that the target remains detectable even when tracking drift occurs. To enhance efficiency, the computational load of the more resource-intensive components of 360° visual tracking must be optimized.

In our previous 360° visual tracking method (last section), each frame used the previous frame's tracking center as the focal point, with the FoV determined based on the target's size in the 2D projection of the previous frame, to perform the 2D gnomonic projection. However, distortion in 360° ERP images is not uniformly distributed across the entire image. It primarily occurs at the left and right edges and near the top and bottom borders—specifically in high-latitude regions and areas close to the horizontal boundaries. To address this, we propose a hybrid approach: applying standard 2D visual tracking methods to the central region of the image while utilizing the 360° visual tracking method for areas near the edges and borders. This approach reduces the computational overhead caused by frequent 2D gnomonic projections and the transformations between 2D and 360° ERP images. For a 360° ERP image with a width of W^{ERP} and height of H^{ERP}, and a target center located at (x_n^{cERP}, y_n^{cERP}), the regions $R(\cdot)$ requiring 360° tracking are defined as follows:

$$\begin{aligned} &R(x_n^{cERP} < \alpha W^{ERP}) \cup R(x_n^{cERP} > (1-\alpha)W^{ERP}) \\ &\cup R(y_n^{cERP} < \beta H^{ERP}) \cup R(y_n^{cERP} > (1-\beta)H^{ERP}) \end{aligned} \tag{4}$$

Where α and β are parameters that control the area in which this algorithm is applied, and they must satisfy $\alpha < 0.5$ and $\beta < 0.5$ to ensure that the entire image area is not included.

In this way, our method reduces the computational load of the 2D gnomonic projection and point transformation in the central region of the image, while enabling targets in this region to utilize the original 2D visual tracking method. By expanding the search area, drifting targets have a higher likelihood of being fully re-tracked, thereby improving robustness.

While this approach improves tracking efficiency and robustness, it introduces a new issue—the scale transition problem between 2D tracking and 360° visual tracking, as well as scale adaptation challenges due to latitude distortion in 360° ERP images. To address this, we update the scale calculation method, replacing the previous direct scaling by latitude with a dynamic adjustment applied to both 2D and 360° tracking regions. The scale change factor for the n-th frame is computed as $\omega_n = \cos(\varphi_n^{cERP}) / \cos(\varphi_{n-1}^{cERP})$, where φ_n^{cERP} and φ_{n-1}^{cERP} denote the target's center latitude in the ERP image for the current and previous frames, respectively. Using this factor, the target's width in the n-th frame is adjusted accordingly. This method compensates for stretching at high latitudes and prevents abrupt scale transitions when switching between tracking modes, ensuring smoother adaptation and more accurate

visual tracking across different regions of 360° ERP images. Then, using ω_n and H_{n-1}^{ERP}, which have been calculated by the tracking method, we can get the width of the target in the n-th frame as $H_n^{ERP} = \omega_n \cdot H_{n-1}^{ERP}$.

Additionally, our algorithm improves Field of View (FoV) selection by expanding its range from fixed values of 60°, 90°, and 120° to six options between 45° and 120° in 15° increments. The adjustment thresholds have been refined from 20 and 400 pixels to 30 and 300 pixels, dynamically decreasing FoV when the target's smaller dimension is below 30 pixels and increasing it when the larger dimension exceeds 300 pixels. During transitions between 2D and 360° visual tracking, the last tracked frame is projected onto a 2D gnomonic image with a 90° FoV, and the target's size determines the FoV for the next frame. This adaptive adjustment enhances tracking accuracy and clarity across varying target sizes.

In general, our algorithm determines whether the target is in the central region of the image based on the tracking box center from the previous frame. At the start of each frame, the algorithm determines whether the tracking box center from the previous frame is within the central region of the image. If the target remains in the central region, the standard 2D visual tracking method is applied, followed by scale adjustment using the computed scale factor. Afterward, the algorithm reassesses whether the target is still in the central region. If so, it proceeds to the next frame without modification. However, if the target moves out of the central region, a 2D gnomonic projection is performed, mapping the tracking box onto a newly generated 2D image, and the feature size is reset accordingly. If the target was already outside the central region in the previous frame, the 360° visual tracking method is used instead. After tracking, scale adjustment is performed to compensate for any distortions. The algorithm then checks whether the target remains outside the central region. If so, another 2D gnomonic projection is performed, and the FoV is adjusted dynamically based on the target's size. If the target has moved back into the central region, the feature size is reset, as the next frame will switch back to the 2D visual tracking method.

By dynamically switching between 2D and 360° visual tracking while adjusting the projection parameters, this method optimizes 360° visual tracking, ensuring smooth transitions and improved tracking efficiency.

3.2.2. Detection of Deformation, Occlusion, Blur and Background Clutter

In the previous section, we discussed ways to make the proposed 360° visual tracking method more robust and efficient. However, several challenges in 360° visual tracking remain unresolved, such as deformation, occlusion, motion blur caused by fast movements, and the tracking of small targets. These issues can lead to tracking failures or drifts, limiting the overall performance of our approach. Moreover, in 360° visual tracking, these challenges tend to occur more frequently, making their resolution a critical aspect of improving the method. Therefore, we will subsequently explore how leveraging knowledge from computer graphics and signal processing can help achieve better tracking results under these complex scenarios.

First, we discuss how to detect deformation, occlusion, blur, and background clutter affecting the target. Here, we adopt the occlusion detection method proposed in (Xu et al. 2022). This approach relies on analyzing the convolution response map generated by the correlation filter. Initially, the target position is identified by examining the peak distribution in the response map. Under normal conditions, a single prominent peak indicates the correct target location. However, when multiple peaks of similar intensity

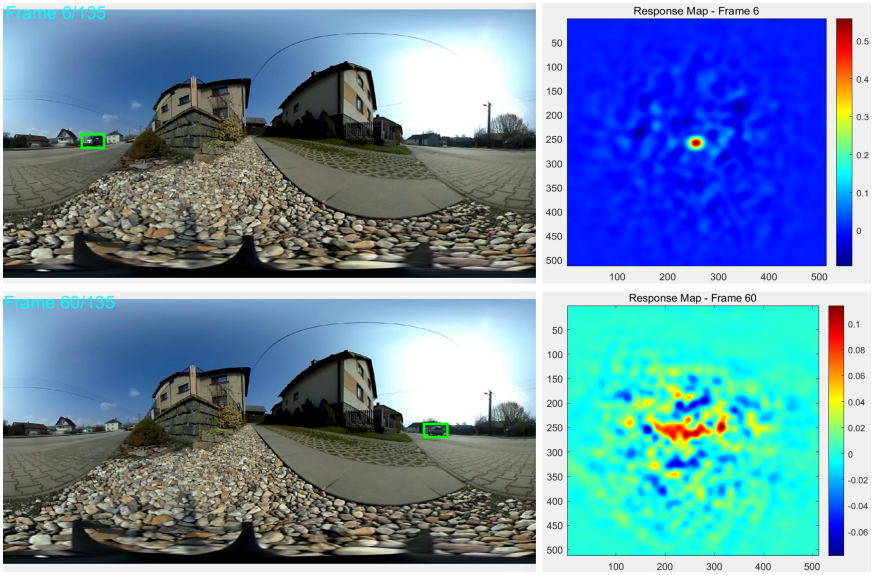

Figure 4. The response graph for general and complex cases.

appear, it may suggest that the target is occluded or affected by interference. To detect such cases, the method defines a threshold to evaluate the intensity difference between the highest and second-highest peaks. If this difference falls below the set threshold, occlusion may be occurring.

While this method is effective for occlusion detection, our experiments indicate that it is a sufficient but not necessary condition for identifying occlusion. Specifically, multiple peaks frequently appear during occlusion, but similar situations also arise when the target undergoes significant deformation or when tracking drift occurs. Therefore, we employ this method for detecting complex tracking scenarios, including occlusion, deformation, motion blur, and tracking drift.

As illustrated in the upper part of Figure 4, when the target is not subject to interference, the response map displays a distinct primary peak with minimal secondary peaks. Conversely, the lower part of Figure 4 demonstrates that under challenging conditions, such as deformation, occlusion, or background interference, multiple peaks emerge, and the intensity of the primary peak decreases significantly. To quantify this phenomenon, the response value at the highest peak is denoted as $Peak_{\text{max}}$. If a nearby secondary reaches at least 80% of $Peak_{\text{max}}$, it is considered a significant secondary peak $Peak_{\text{side}}$. Here, we can find that if the complex tracking situations happen, the number of $Peak_{\text{side}}$ is more than 2 or the value of $Peak_{\text{max}}$ is less than 0.1. These values are determined by a lot of experiments and parameters. By adopting this method, various challenging tracking scenarios can be identified effectively, enabling the tracker to adapt and maintain robust performance.

3.2.3. Sample Set Method

Upon detecting challenging scenarios such as deformation, occlusion, blur, and background clutter, appropriate tracking adjustments are required. As the current approach employs a DCF-based visual tracking method, erroneous filter updates may degrade performance in these cases. To mitigate this, maintaining a sample set that stores filter information from frames with reliable tracking, as proposed in (Huang et al. 2024), can enhance robustness. These stored samples assist in tracking when encountering

complex conditions, preventing the filter from learning incorrect data.

To implement this, frames affected by occlusion, deformation, background clutter, or motion blur are identified, and specific measures are applied. First, in the correlation filter-based method, the filter typically updates at fixed intervals. However, if a complex tracking condition is detected, the update is skipped for that frame. Additionally, similar to (Huang et al. 2024), filters used during tracking are stored in a sample set, though without employing a full PN-tree structure. Instead, the sample set aids in recovering targets under difficult conditions.

Then, frames are categorized based on whether the previous frame's tracking was performed using the original 2D visual tracking method or the 360° visual tracking method. For targets in the image's central region, the existing UPDT filter is sufficient, as it already handles typical challenges like deformation in conventional videos. Since distortion is less prominent in central regions of 360° ERP images, additional modifications are unnecessary. However, in frames where the original 2D tracking method is applied, if complex tracking conditions are detected, the filter update is skipped to prevent contamination. In contrast, if the target's center position in the previous frame is not within the central region of the 360° ERP image, the filter update for that frame is also skipped.

If no complex tracking situation is detected in the current frame using the method in last subsection, a sample is stored; otherwise, the sample set is utilized for tracking. In the storage process, the target position search remains unchanged, while sample storage follows a specific logic. A maximum of five samples can be stored at any time. For the first frame using the 360° visual tracking method, if no complex tracking condition is detected, the filter trained on the target features is stored as the first sample. Subsequently, each updated filter is stored until the five slots are filled, with new samples replacing older ones in a first-in, first-out (FIFO) manner.

Once the sample set reaches five filters, each filter update requires computing its correlation with stored samples. If the lowest correlation value falls below a predefined threshold, the current filter replaces the oldest sample. This process continues throughout the video. When a frame is identified as having complex tracking conditions, its filter update is skipped, if applicable, and the sample set is used instead. If the sample set is empty, the current filter is applied for target position search. Otherwise, correlation values between the current filter and stored samples are computed, and the least correlated filter is selected for tracking.

This sample set approach prevents filter contamination by unreliable frames, ensuring stable and accurate tracking across the video.

3.2.4. Target Prediction in More Specific Cases

To mitigate the impact of deformation, occlusion, and blurring on tracking results, this method detects target positions in frames identified as complex tracking scenarios and integrates the results into prediction. Inspired by (Xu et al. 2022) and (Wang et al. 2017), a constrained Kalman filter predicts the target's position and dimensions in each frame, incorporating preprocessing for boundary crossings, weighted regression for trend estimation, and outlier removal for robustness.

The tracking results from the past 10 frames serve as the basis for prediction, with all available frames used if fewer than 10 exist. Given the boundary characteristics of 360° images, x-coordinates are adjusted when a frame-to-frame shift exceeds half the image width $W^{ERP}/2$ to prevent discontinuities. Outliers in the tracking results are detected by computing their deviation from the mean displacement and are replaced

by interpolated values when necessary.

A weighted regression model is then employed to predict coordinates using a time-indexed linear regression equation $y_t = \beta_0 + \beta_1 t, \quad t = 1, 2, \ldots, n_{\max}$, where β_0 and β_1 are estimated through Gaussian-weighted least squares. To maintain consistency with past motion patterns, trend adjustments are applied, and abrupt changes are constrained by setting a maximum displacement threshold. This threshold is defined as 1.3 times the mean of the absolute sum of coordinate variations over the past $n_{\max}$ frames, ensuring smoother transitions in motion predictions.

To maintain continuity in 360° images, predicted x-coordinates are mapped within valid image boundaries using modulo operations. The predicted target width and height are estimated based on historical trends:

$$W_n^{ERP} = W_{n-1}^{ERP} + \sum_{i=n-n_{\max}}^{n-1} \Delta W_i^{ERP}, \quad H_n^{ERP} = H_{n-1}^{ERP} + 0.5 \times \sum_{i=n-n_{\max}}^{n-1} \Delta H_i^{ERP} \tag{5}$$

By integrating these steps, the method enhances tracking accuracy in challenging conditions while ensuring smooth transitions and reliable scale adaptation across frames.

3.2.5. Decision Fusion for Target Position and Scale Estimation

The current frame's coordinate and scale results are obtained from two sources: the sample set method and the target prediction. Determining the final values requires an adaptive weighting approach based on tracking confidence.

First, the response map from the sample set method is re-evaluated using the metrics described before, which is the highest peak response value $Peak_{\max}$ and the number of significant secondary peaks $Number(Peak_{\text{side}})$. A lower $Peak_{\max}$ or a higher $Number(Peak_{\text{side}})$ indicates better tracking accuracy. Thus, the confidence weight for the sample set result is defined as:

$$\begin{aligned} \omega_{\text{sampleset}} = 0.65 \cdot & \frac{1}{1 + \exp\left((Peak_{\max} - 0.16) \cdot 10\right)} \\ + 0.35 \cdot & \frac{1}{1 + \exp\left((Number(Peak_{\text{side}}) - 2) \cdot 1.5\right)} \end{aligned} \tag{6}$$

The weight for the Kalman filter prediction is complementary: $\omega_{\text{predicted}} = 1 - \omega_{\text{sampleset}}$. The final coordinates are computed as a weighted sum:

$$\begin{cases} x_n^{ERP} = \omega_{\text{sampleset}} \cdot x_{n\_sampleset}^{ERP} + \omega_{\text{predicted}} \cdot x_{n\_predicted}^{ERP} \\ y_n^{ERP} = \omega_{\text{sampleset}} \cdot y_{n\_sampleset}^{ERP} + \omega_{\text{predicted}} \cdot y_{n\_predicted}^{ERP} \end{cases} \tag{7}$$

Similarly, for scale estimation, the width and height from both methods are weighted using the same sigmoid-based confidence measure. The final width and height are determined in the same manner as the coordinates.

By adaptively balancing the sample set and target predictions based on response map confidence, this method enhances tracking robustness and ensures smooth transitions in complex scenarios.

4. Experiments and Results

To demonstrate the superiority and effectiveness of this method in 360° visual tracking, experiments were conducted on the 360° visual tracking dataset using the improved UPDT360 method described in last chapter. Additionally, for the method in last chapter that combines conventional 2D visual tracking with 360° visual tracking, a parametric sensitivity analysis was performed to evaluate the impact of the region size used for 360° visual tracking. Lastly, the performance of our improved UPDT360 method was compared against the state-of-the-art SAM2 method, which is based on segmentation tracking, to investigate their respective strengths and weaknesses. All experiments were implemented in MATLAB 2023b and conducted on a PC equipped with an Intel Core i7-14650HX CPU, 16 GB RAM, and a single NVIDIA GTX 4050 GPU.

4.1. Experimental Datasets

In this section, we evaluate the performance of our proposed 360° visual tracking method using three trackers across various datasets. The primary dataset is a 360° ERP video benchmark in the OTB format (Wu et al. 2013), which includes videos from (Ambrož 2024; Mi and Yang 2019; Liu et al. 2018; Nasrabadi et al. 2019). This dataset comprises 21 challenging sequences that encompass scenarios such as occlusion, deformation, viewpoint changes, and fast motion, alongside challenges unique to 360° ERP videos, including distortion, boundary artifacts, and stretching near the poles.

Furthermore, we conducted experiments on the 360VOT dataset (Huang et al. 2023), a recent benchmark specifically designed for omnidirectional tracking. This dataset contains 120 sequences spanning 32 categories and introduces new evaluation metrics, such as dual success rate and angle precision. It provides additional validation of our method's capability to handle complex omnidirectional challenges effectively.

4.2. Comparison Methods

In experiments in (Peng and Zhang 2024), we selected three existing filters (STRCF (Li et al. 2018), DeepSTRCF (Li et al. 2018), and UPDT (Bhat et al. 2018)), and continued using the previously improved 360° tracking methods based on these filters: STRCF360, DeepSTRCF360, and UPDT360. Additionally, we incorporated ASRCF (Dai et al. 2019) and its improved ASRCF360 filter in 360° visual tracking. This approach learns object-specific adaptive spatial weights that dynamically adjust to appearance variations. Furthermore, several popular correlation filter-based tracking methods from recent years were included in the comparison. These methods include LADCF (Xu et al. 2019b), which ranked first in the VOT2018 challenge; ECO (Danelljan et al. 2017), renowned for its balance of efficiency and accuracy, and its handcrafted-feature variant ECO-HC (Danelljan et al. 2017); and GFSDCF (Xu et al. 2019a), which significantly improves tracking accuracy while reducing feature redundancy and efficiently implementing tracking. We conducted a comprehensive comparison of these DCF-based 2D visual tracking methods, along with four 360° tracking methods derived from them, against our proposed method in this chapter. The comparison focuses on evaluating the tracking success rate and precision to validate the advantages of our approach.

4.3. Parametric Sensitivity Analysis

In this section, we perform a parameter sensitivity analysis on the method proposed in last chapter, which employs the original 2D visual tracking UPDT method in the central region of the image and the improved UPDT360 method in the boundary regions of 360° ERP videos. This analysis aims to identify which parts of the image should be treated as boundary regions and which as central regions to achieve optimal tracking performance.

To determine the optimal boundary configuration for 360° and 2D tracking integration, we analyze the left-right and top-bottom boundaries separately. Since left-right boundaries primarily involve edge connections with minimal distortion, the parameter range is set between 0.1 and 0.4. If denoted as s_1, 360° tracking is applied when $0 < x_n^{ERP} < s_1 \cdot W^{ERP}$ and $(1 - s_1) \cdot W^{ERP} < x_n^{ERP} < W^{ERP}$, while 2D tracking is used in the central region.

For the top-bottom boundaries, high-latitude distortions require a narrower range of 0.2 to 0.4. When represented as s_2, 360° tracking is applied within $0 < y_n^{ERP} < s_2 \cdot H^{ERP}$ and $(1 - s_2) \cdot H^{ERP} < y_n^{ERP} < H^{ERP}$, while 2D tracking is utilized in the central region.

Experiments on the first 360° video dataset evaluate these parameters. Initially, the top-bottom boundary is fixed at 0.4, and the left-right boundary varies between 0.1 and 0.4 (Figure 5). Results indicate that while 0.1 achieves the highest precision, it has the lowest success rate. The best overall performance is achieved at 0.3, leading to its selection as the left-right boundary parameter.

Next, fixing the left-right boundary at 0.3, the top-bottom boundary is adjusted between 0.2 and 0.4 (Figure 6). Although 0.2 provides better precision, 0.4 yields the highest success rate, making it the final choice.

Therefore, the optimal boundary configuration is defined as $0 < x_n^{ERP} < 0.3 \cdot W^{ERP}$ and $0.7 \cdot W^{ERP} < x_n^{ERP} < W^{ERP}$, or $0 < y_n^{ERP} < 0.4 \cdot H^{ERP}$ and $0.6 \cdot H^{ERP} < y_n^{ERP} < H^{ERP}$, where the UPDT360 method is applied. Conversely, in the central region, $0.4 \cdot W^{ERP} < x_n^{ERP} < 0.6 \cdot W^{ERP}$ or $0.3 \cdot H^{ERP} < y_n^{ERP} < 0.7 \cdot H^{ERP}$, the UPDT method is used, ensuring optimal tracking performance.

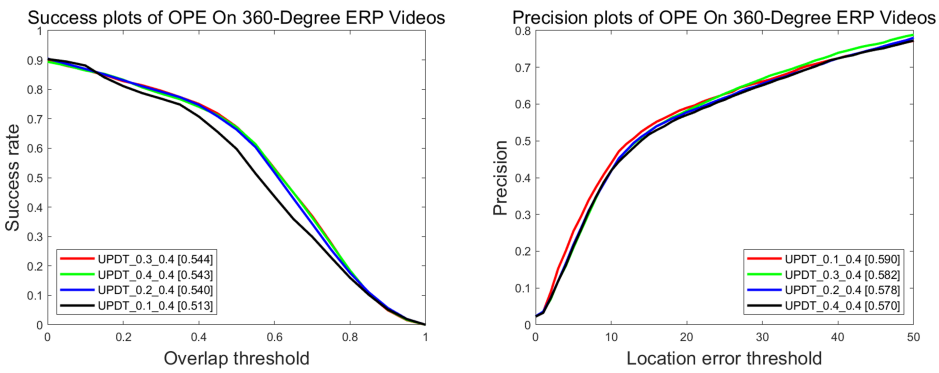

Figure 5. The result of sensitivity analysis for x-coordinates.

4.4. Quantitative Analysis for the 360OTB Dataset

In this session, we will conduct a quantitative analysis by combining the improved method proposed in last chapter with the improvements introduced in last section

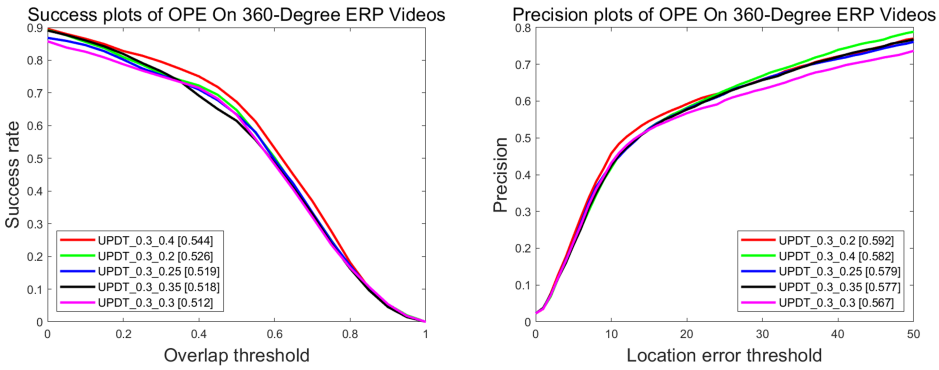

Figure 6. The result of sensitivity analysis for y-coordinates.

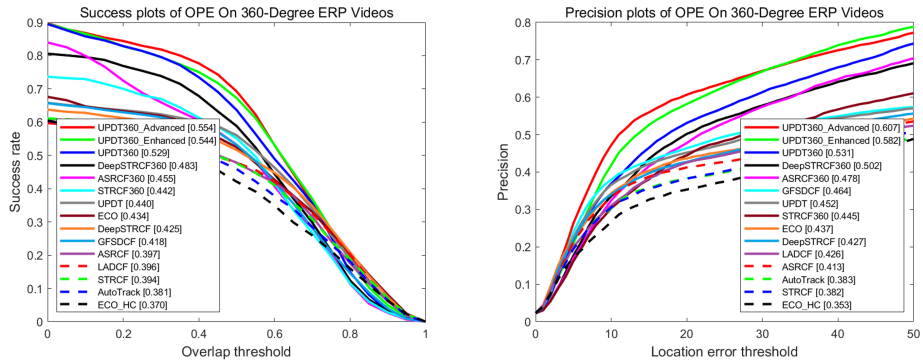

Figure 7. Precision and success rate plots on the 360OTB dataset.

and applying them to 360OTB dataset.

Similar to (Peng and Zhang 2024), we name the traditional UPDT tracking method, which has been improved using the 360° tracking approach, as UPDT360. We name the tracking results of combining UPDT360 with the method proposed in last section, which integrates traditional tracking with 360° tracking, as UPDT360_Enhanced. Based on this, the tracking results that incorporate the handling of complex tracking scenarios are named UPDT360_Advanced. Using these as a foundation, along with the 360° tracking algorithms improved in the previous section, their original 2D tracking methods, and the comparative methods described before, experiments are conducted on the first dataset. Figure 7 illustrates the comparison results between UPDT360_Enhanced, UPDT360_Advanced, UPDT360 and other methods.

In Figure 7, it is evident that UPDT360 significantly outperforms UPDT and other 2D tracking methods. The success rate and precision scores of UPDT360 demonstrate its superior ability to handle the challenges posed by 360° ERP videos, such as distortions and viewpoint changes. The UPDT360_Enhanced method further improves the success rate and precision by 1.5% and 5.1%, respectively, over the original UPDT360. The notable improvement in precision is primarily due to a more concentrated tracking response, which facilitates higher localization accuracy using standard 2D visual tracking approaches. Meanwhile, the success rate benefits from the newly introduced scale calculation, ensuring better target coverage within the tracking box. The UPDT360_Advanced method achieves the highest overall performance in both success rate and precision. Compared to UPDT360, it increases the success rate by

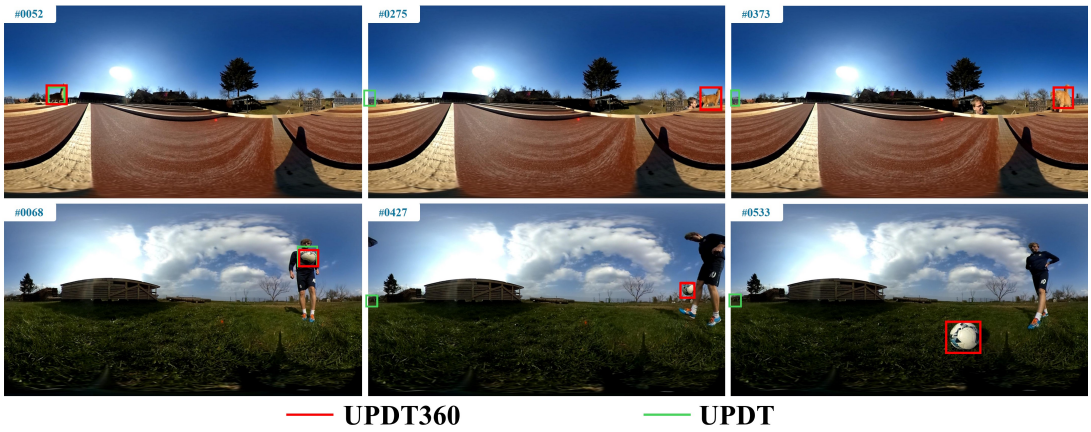

Figure 8. Tracking results of UPDT and UPDT360. The red box represents the tracking results of UPDT360, while the green box indicates the results of UPDT.

2.5% and improves precision by 7.6%, surpassing UPDT360_Enhanced by an even larger margin. These results confirm that the enhancements proposed in last chapter effectively contribute to improved 360° visual tracking performance.

Figure 8 compares the tracking results of UPDT and UPDT360 on two 360° ERP videos, highlighting both traditional tracking challenges and unique 360° difficulties. In the first video (top row), the target encounters illumination changes, boundary crossing, and deformation due to varying viewing angles. While UPDT adapts well to illumination changes, it struggles with 360°-specific challenges. In contrast, UPDT360 effectively handles these issues, ensuring more precise tracking. In the second video (bottom row), the target, a soccer ball, faces rapid motion, boundary crossing, and width elongation from high-latitude distortion. UPDT fails to handle these combined challenges, whereas UPDT360 successfully addresses them, maintaining accurate and robust tracking throughout the video.

Figure 9 compares the tracking results of UPDT360_Advanced, UPDT360_Enhanced, and UPDT360 across three different videos. In the first video (top), the target is heavily affected by background clutter. UPDT360_Advanced shows a clear advantage in tracking position and scale, while UPDT360_Enhanced also improves upon the original UPDT360. The second video (middle) presents motion blur and occlusion challenges. UPDT360_Advanced performs significantly better, accurately tracking the target during blurred frames and successfully re-identifying it after occlusion. The third video (bottom) involves target deformation. Both UPDT360_Advanced and UPDT360_Enhanced demonstrate better scale adaptation than UPDT360, with UPDT360_Advanced excelling further in handling scale variations, ensuring more stable tracking performance.

Table 1 presents the overall FPS (Frames Per Second) for the three methods. From it, we can conclude that in terms of tracking speed, our methods also offer certain advantages over UPDT360. Specifically, the UPDT360_Enhanced method demonstrates faster performance compared to the original UPDT360. This is primarily because the frequency of performing 2D gnomonic projections has been reduced. Unlike the original approach, the improved method does not require updates on every frame, thereby decreasing the overall computational load. While the UPDT360_Advanced method is slightly slower than UPDT360_Enhanced, this is due to the additional logic introduced in last chapter to handle complex tracking scenarios. However, it is still faster than

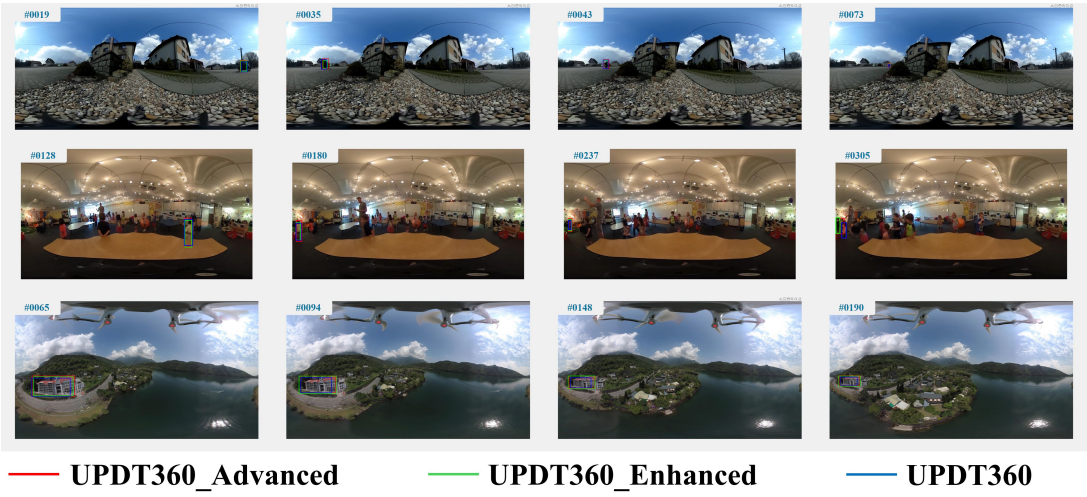

Figure 9. Tracking results of UPDT360_Advanced, UPDT360_Enhanced, and UPDT360. The red box represents the tracking results of UPDT360_Advanced, the green box indicates the results of UPDT360_Enhanced, and the blue box corresponds to the results of UPDT360.

Table 1. Overall FPS of Our Trackers

Tracker	UPDT360	UPDT360_Enhanced	UPDT360_Advanced
FPS	0.7744	0.9120	0.8069

the original UPDT360. These results indicate that our improved methods not only enhance tracking accuracy but also achieve better tracking efficiency.

4.5. Quantitative Analysis for the 360VOT Dataset

Figure 10 presents the overall comparison results of our methods with other approaches on the 360VOT dataset.

Although the UPDT360 method proposed in the previous chapter does not show significant advantages over the original UPDT method, our UPDT360_Enhanced and UPDT360_Advanced methods achieve notable improvements. Among all traditional 2D tracking and 360° tracking methods, UPDT360_Enhanced and UPDT360_Advanced rank in the top two. Specifically, UPDT360_Advanced and UPDT360_Enhanced improve the success rate by 1.6% and 2.9%, respectively, compared to UPDT360. Their precision rates increase by 0.7% and 2.5%, respectively, relative to UPDT360.

The 360VOT dataset, unlike the 360OTB dataset, contains more videos and presents greater tracking challenges, frequently involving various complex tracking scenarios. These results demonstrate that when confronted with such challenges, our methods consistently achieve better success rates and precision, underscoring their effectiveness in handling complex tracking situations.

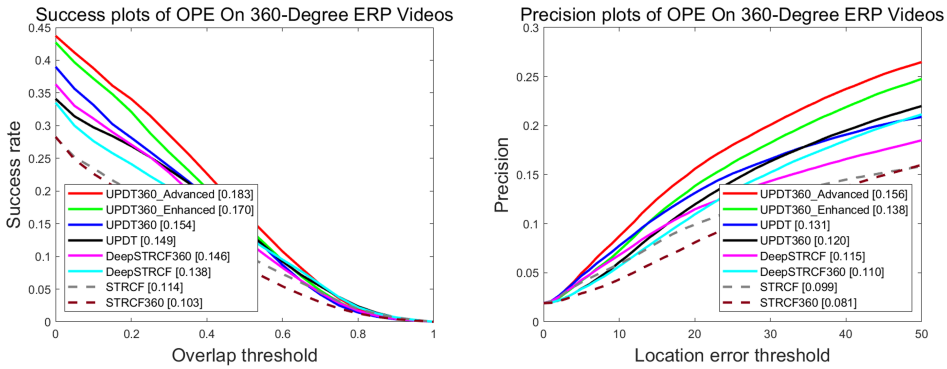

Figure 10. Precision and success rate plots of trackers on 360VOT dataset.

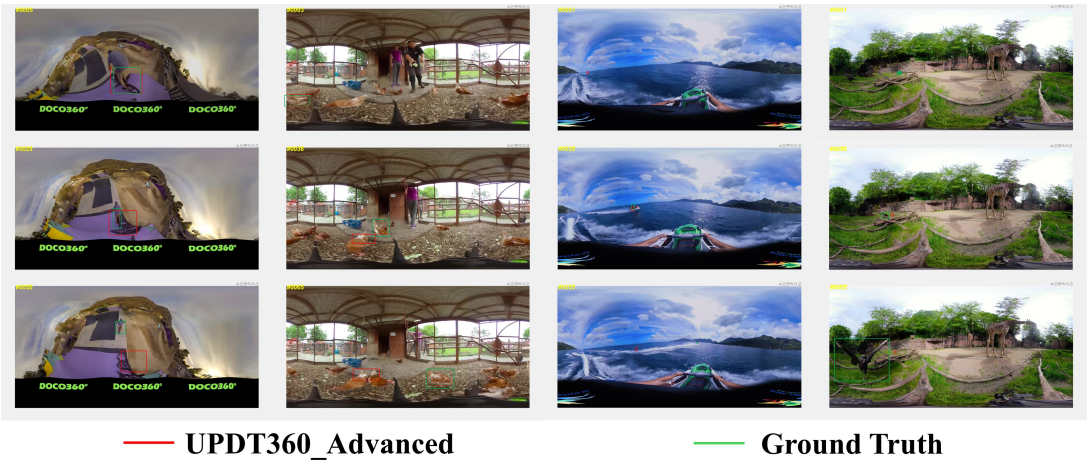

Figure 11. Analysis of several failure cases

4.6. Failure case analysis

However, since we adopt a DCF-based framework as the foundation of our tracking method, it inherently relies on previously extracted features and the trained filters. At the same time, our approach cannot handle all complex scenarios. Therefore, some tracking failures still occur. Here, we analyze several representative failure cases.

Due to our use of a fixed rectangular tracking box, the model learns from the entire content within the rectangle during each update. If the target object is irregularly shaped or occupies only a small portion of the bounding box, as in the first column of Figure 11 (the bicycle), the filter tends to learn a large amount of background information, which leads to tracking failure.

As shown in the second column of Figure 11, the video contains many objects that resemble the target. In such cases, if the target continues to move, causing occlusion or overlap with similar-looking objects, it becomes easy for the tracker to mistakenly follow a different object, resulting in failure.

In the third column of Figure 11, the target object is small, moves quickly, and experiences partial occlusion. These factors easily cause tracking drift. Additionally, rapid motion in 360-degree videos often results in irregular movement trajectories, which further increases the likelihood of tracking failure.

Tracking in DCF-based frameworks depends heavily on feature extraction, particularly from the initial frames. As shown in the fourth column of Figure 11, the target

object appears highly unclear and visually ambiguous in the early frames. Under such conditions, it becomes difficult to extract sufficient features for learning the target's appearance, ultimately leading to failure.

These cases illustrate common types of tracking failure. Overall, our method still struggles in scenarios involving background clutter, similar-looking objects, fast motion, small targets, and unclear initial features. These limitations indicate areas for potential improvement.

4.7. Comparing with the SAM2

Currently, segmentation-based methods like SAM2 (Ravi et al. 2024) are among the most popular approaches in tracking. To evaluate and compare the tracking accuracy and efficiency of SAM2 with our methods, we ran SAM2 on videos from our dataset. However, due to limited GPU memory, processing longer videos with SAM2 was not feasible. Instead, we selected a subset of representative videos for comparison. These videos are identified by their video numbers, where numerical values correspond to videos from the first 360° video dataset, while IDs starting with "VOT" indicate videos from the 360VOT dataset. Each selected video includes at least one of the following tracking challenges: cross-border movement, latitude variation, or traditional 2D tracking challenges including deformation and occlusion. Videos with the cross-border attribute are labeled as CB, those affected by latitude variation are labeled as LV, videos exhibiting deformation are marked as DE, and those containing occlusion are labeled as OC. Since VOT videos are generally longer and encompass multiple challenges, isolating the effect of each attribute on tracking performance is difficult. Consequently, we selected relatively fewer videos from this dataset. Table 2 presents a comparison of tracking accuracy between our UPDT360_Advanced method and SAM2 on these selected videos, with the best results for each video highlighted in red.

Table 2. Tracking Accuracy Compared with SAM2

Video ID	**Attributes**	**Accuracy**		
		UPDT360	**UPDT360_Advanced**	**SAM2**
02	CB, LV, DE	0.62678	0.63652	0.30094
05	DE	0.59746	0.66148	0.71313
14	CB, LV, DE	0.28078	0.26984	0.14109
15	CB	0.26988	0.51814	0.23469
17	DE, OC	0.42817	0.50365	0.85993
20	OC	0.46704	0.46515	0.70023
21	CB, LV, DE	0.23768	0.24546	0.26287
VOT03	OC, DE	0.02381	0.02705	0.22181
VOT04	CB, LV, DE, OC	0.01118	0.00821	0.00464
VOT57	CB, DE	0.46582	0.57794	0.38778

From the results in Table 2, it is evident that our method significantly outperforms SAM2 in videos with the cross-border attribute while achieving comparable performance in those with the latitude variant attribute. However, SAM2 performs better in handling deformation and occlusion. This discrepancy arises because SAM2 loses track of objects when they cross the left or right boundaries, whereas it excels at

Table 3. Tracking FPS Compared with SAM2

Tracker	UPDT360	UPDT360_Enhanced	UPDT360_Advanced	SAM2
FPS	0.7744	0.9120	0.8069	0.0575

handling latitude variations, deformation, and occlusion by effectively extracting and retaining object boundary features. These findings indicate that while our method is less effective in boundary extraction compared to SAM2, it offers a clear advantage in addressing boundary-related challenges unique to 360° videos.

Table 3 presents a comparison of tracking speed, measured in FPS, between our method and SAM2. The results indicate that our method has a clear advantage in tracking speed. SAM2 is significantly slower, making it unsuitable for online tracking and lagging far behind in tracking efficiency.

In summary, while our method is less effective than SAM2 in handling some traditional tracking challenges, it demonstrates notable advantages over SAM2 in addressing boundary issues unique to 360° videos and in tracking speed.

5. Application in 360° Video Editing

While 360° visual tracking has valuable applications in areas such as video surveillance and intelligent transportation, its potential to significantly enhance the user experience in the VR field lies in its integration with 360° video editing technologies. 360° video editing involves modifying dynamic objects while maintaining spatial and temporal consistency. Traditional methods often rely on optical flow or manual segmentation masks, which can be prone to errors due to geometric distortions in equirectangular projections. To address these challenges, we integrate our 360° visual tracking method with existing editing frameworks, enabling automatic object tracking and modification across frames.

Our approach consists of two main editing strategies: one based on the Neural Panoramic Representation (NPR) framework (Kou et al. 2024) and another extending the Segment Anything Model for video (SAM2) (Ravi et al. 2024). In the NPR-based editing, UPDT360 tracking results replace manually provided masks, allowing automatic identification and selection of target objects. This eliminates the need for frame-by-frame segmentation, improving flexibility and efficiency. Beyond NPR-based editing, we further enhance 360° video editing by integrating UPDT360 with SAM2. While SAM2 enables object segmentation and editing through user prompts, it lacks an inherent understanding of 360° boundary continuity. Our method addresses this limitation by leveraging UPDT360 to maintain tracking consistency when objects cross the left or right frame edges in equirectangular videos, ensuring seamless editing.

In this section, we introduce three specific editing operations under these two frameworks, demonstrating how our tracking-enhanced approach improves accuracy and efficiency in 360° video object editing.

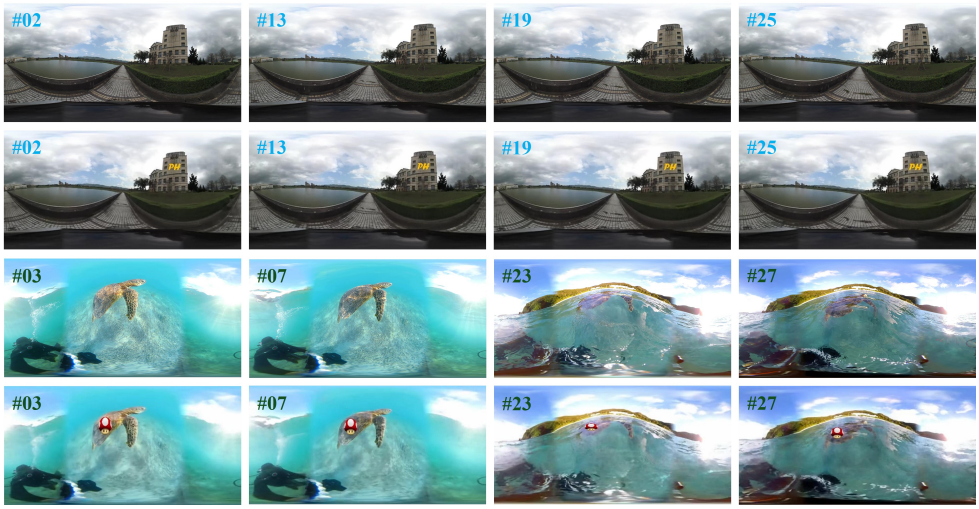

Figure 12. Results of moving object geometric-aware editing. This figure consists of four rows: the first two rows correspond to the editing results of the first video (Building), while the last two rows correspond to the second (Turtle). For each video, the first row shows the original frames, and the second presents the edited results. Frame indices are provided to indicate the temporal position within the video.

5.1. Moving Object Geometric-aware Editing

Our first editing application involves geometric-aware editing on moving objects, enabling the addition of patterns or text that remain attached to the object, moving and deforming along with it. By integrating NPR with our UPDT360 tracking method, we achieve seamless texture mapping on moving objects. Due to GPU memory constraints and NPR performance limitations, each edited video contains only 30 representative frames. The specific implementation results are shown in Figure 11.

In Figure 11, we apply geometric-aware editing to a building (top) and a turtle (bottom). The top part of Figure 11 presents a video from 360OTB dataset. In each set of results, the first row shows the original video frames, while the second rows present the edited results. Despite deformations and motion caused by camera movement, our method successfully overlays the letters "PH" in two different colors and fonts onto the building's surface, naturally blending them like graffiti. The bottom part of Figure 11 presents a video from the 360VOT (Huang et al. 2023) dataset, introducing greater challenges in tracking and editing moving objects compared to the previous video. The difficulty arises from the camera's transition from underwater to above water, causing the turtle to undergo shape deformation. Additionally, after emerging above water, the turtle's outline becomes less distinct due to partial occlusions from water reflections. In this scenario, extracting a foreground mask for each frame using existing methods is highly challenging due to the dynamic nature of both the scene and the object. However, by first tracking the target's position with our 360° visual tracking method and then applying NPR for editing, we achieve more stable results. Even during the transition from underwater to above water, our method maintains accurate tracking and enables successful object editing.

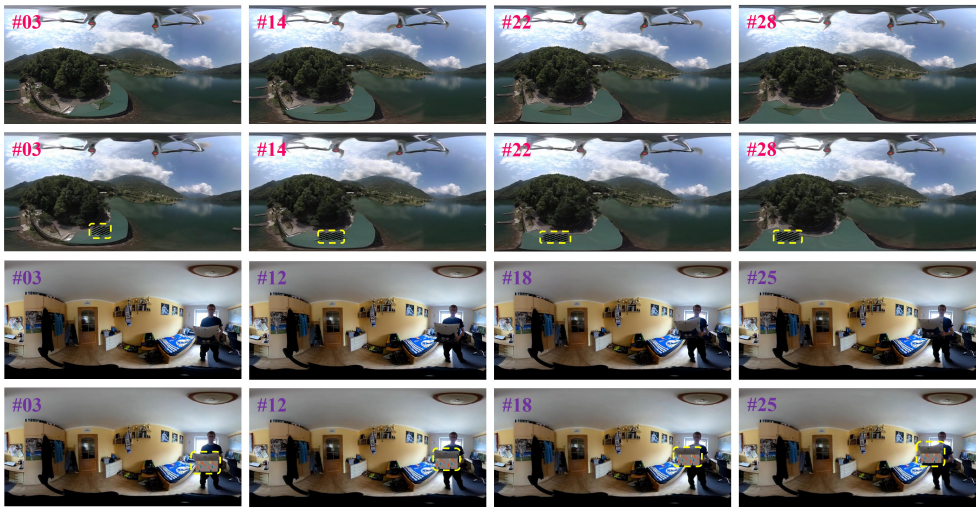

Figure 13. Results of moving object texture replacement. This figure consists of four rows: the first two rows show the editing results of a video where a mosaic effect is applied to the roof, while the last two rows show the results of another video where the texture of the pillow is replaced. In each video, the first row presents the original frames, and the second displays the edited results. Frame indices are provided to indicate the temporal position within the video.

5.2. Moving Object Texture Replacement

Our second editing application involves texture transformation on objects, where the entire texture of a moving object is replaced with an edited version or precisely censored using mosaics. The specific implementation results are shown in Figure 12.

In Figure 12, we apply full texture replacement to a rooftop (top) and a pillow (bottom), both sourced from the 360OTB (Ambrož 2024; Mi and Yang 2019; Liu et al. 2018; Nasrabadi et al. 2019) dataset. In the top part of Figure 12, the rooftop is fully covered with a mosaic effect. By leveraging tracking for precise localization, we successfully apply texture replacement to ensure that the mosaic pattern fully covers the rooftop while leaving surrounding objects unaffected. In the bottom part of Figure 12, the original pillow, which features an animal face design, is replaced with a plain pillow decorated with multicolored star patterns. The new texture seamlessly follows the pillow's movement and deformation. To enhance clarity, we highlight the texture replacement target areas with yellow bounding boxes in the edited images. Integrating our method with NPR enables precise texture replacement for moving objects in 360° videos, ensuring seamless and realistic modifications.

5.3. Moving Object Boundary Connection

In the previous discussion, we introduced how (Ravi et al. 2024) employs object boundary extraction for visual tracking. Beyond tracking, SAM2 enables basic editing of tracked objects using promptable visual segmentation (PVS) and interactive video object segmentation (iVOS). With user prompts like clicks, bounding boxes, or masks, SAM2 tracks objects across frames using a streaming memory mechanism, ensuring segmentation consistency. Once tracking is complete, it determines object boundaries and applies color modifications, either through random mappings, color jittering, or local style transfer, preserving background integrity. The final adjustments are mapped

Figure 14. Results of moving object boundary connection. This figure consists of four rows: the first two rows show the editing results of the first video (Cat), while the last two rows correspond to the second (Dancing). In each video, the first row presents the results edited using SAM2 alone, while the second row shows the results after combining SAM2 with our 360° visual tracking method. Frame indices are provided to indicate the temporal position within the video.

back onto the 360° ERP video, ensuring frame consistency.

However, as analyzed in last chapter, our comparison shows that SAM2 struggles when objects cross the left or right boundaries of the frame. Since SAM2 was not designed for 360° ERP videos, it lacks an understanding of boundary connections, leading to tracking failures when objects transition across frame edges.

To address it, our third application extends the original SAM2 method by integrating our 360° visual tracking approach. Specifically, when an object partially overlaps with the left or right boundary, our tracking algorithm replaces the SAM2 output, ensuring seamless boundary continuity. This enhancement enables consistent object tracking and editing across 360° ERP videos. Figures 13 illustrate the results of editing tracked objects by randomly altering their colors in two different video sequences. The first set (top) of results is generated using only SAM2, while the second set (bottom) combines SAM2 with our UPDT360 method. These experiments were conducted on videos from both the 360OTB dataset (Ambrož 2024; Mi and Yang 2019; Liu et al. 2018; Nasrabadi et al. 2019) and the 360VOT dataset (Huang et al. 2023).

In the top part of Figure 13, when the cat crosses the boundary, SAM2 (1st row) fails to maintain consistent editing, causing the target to disappear in the output. In contrast, integrating SAM2 with UPDT360 (2nd row) ensures continuous object editing. A similar issue occurs in another video results in Figure 13, where the dancing person crosses the boundary. SAM2 (3rd row) partially detects and edits the target, but the result is incomplete and unstable, struggling with accurate tracking after the transition. Instead, utilizing SAM2 in combination with UPDT360 (4th row) ensures seamless editing before and after crossing, though during the transition, the target is sometimes only partially edited, leaving room for improvement. However, once the object is fully visible again, tracking and editing remain consistent.

Therefore, by integrating SAM2 with our 360° visual tracking approach, we effectively enhance editing performance in 360° videos, addressing the challenges posed by boundary transitions and ensuring a more stable and accurate object editing process.

Figure 15. One results with all three moving object editing methods. The first row shows the original frames of the video, while the second presents the edited results. Frame indices are provided to indicate the temporal position within the video.

Finally, Figure 14 demonstrates all three moving object editing methods integrated with our 360° visual tracking approach. The Moving Object Geometric-aware Editing method modifies the content on the right-side billboard, while the Moving Object Texture Replacement method changes the color of the taxi in the center. Additionally, the Moving Object Boundary Connection method applies a mosaic effect to the person on the left-side billboard, enabling the editing of multiple moving objects within the scene. For better visualization, we use yellow bounding boxes in the edited images (the second row) to indicate the specific areas where texture replacement has been applied.

These results highlight the versatility of our 360° visual tracking method, showcasing its ability to support various moving object editing applications in 360° videos. This enhances user interactivity and engagement in VR video experiences.

6. Conclusion

In this paper, we first present a method to convert 360° visual tracking into 2D visual tracking via projection. Based on this approach, we further explore strategies to enhance tracking accuracy, robustness, and efficiency. To improve precision and computational efficiency, we classify the image into central and boundary regions, dynamically adjust the scale based on latitude, and adapt the FoV according to the target's size.

To further enhance performance, we introduce a sample set mechanism to detect frames where tracking quality deteriorates. Additionally, we employ a Kalman Filter-based trajectory prediction method to estimate the target's position and size in frames where tracking fails. This mechanism complements the sample set approach, achieving more accurate and stable tracking in challenging conditions.

Experiments conducted on two datasets validate the effectiveness of our proposed methods, demonstrating notable improvements in tracking precision and success rates while ensuring adaptability to various 360° ERP attributes. We further integrate the proposed tracking method into an existing 360° editing application, verifying its practical applicability.

While our method is not inherently designed for multi-object tracking, multiple instances of the same tracking framework can be executed in parallel to track multiple targets. However, this approach increases computational complexity proportionally to the number of targets, as each requires an independent feature set and tracking process. Regarding editing, it is feasible to simultaneously edit multiple targets as long as there is no occlusion between them. In scenarios involving occlusion, additional processing is needed to distinguish and handle overlapping targets effectively.

Future improvements could include integrating Transformer-based tracking for enhanced accuracy and adapting bounding boxes to better fit 360° ERP images to reduce tracking drift. Additionally, leveraging deep learning for 360° feature extraction presents promising opportunities for further advancements in 360° visual tracking.

Acknowledgment

This work was supported by the Marsden Fund Council managed by the Royal Society of New Zealand (No. MFP-20-VUW-180), and by the Faculty Strategic Research Grant of Victoria University of Wellington (Project No. 412684).

References

Ambrož V. 2024. 360tracking: A benchmark for 360-degree visual object tracking; [https://github.com/VitaAmbroz/360Tracking?tab=readme-ov-file]. Accessed: 2024-08-11.

Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr PH. 2016. Staple: Complementary learners for real-time tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 1401–1409.

Bhat G, Danelljan M, Gool LV, Timofte R. 2019. Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF international conference on computer vision. p. 6182–6191.

Bhat G, Johnander J, Danelljan M, Khan FS, Felsberg M. 2018. Unveiling the power of deep tracking. In: Proceedings of the European conference on computer vision (ECCV). p. 483–498.

Bolme DS, Beveridge JR, Draper BA, Lui YM. 2010. Visual object tracking using adaptive correlation filters. In: 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE. p. 2544–2550.

Cai C, Liang X, Wang B, Cui Y, Yan Y. 2018. A target tracking method based on kcf for omnidirectional vision. In: 2018 37th Chinese Control Conference (CCC). IEEE. p. 2674–2679.

Chen X, Yan B, Zhu J, Wang D, Yang X, Lu H. 2021. Transformer tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p. 8126–8135.

Coors B, Condurache AP, Geiger A. 2018. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the European conference on computer vision (ECCV). p. 518–533.

da Silveira TL, Pinto PG, Murrugarra-Llerena J, Jung CR. 2022. 3d scene geometry estimation from 360 imagery: A survey. ACM Computing Surveys. 55(4):1–39.

Dai K, Wang D, Lu H, Sun C, Li J. 2019. Visual tracking via adaptive spatially-regularized correlation filters. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p. 4670–4679.

Danelljan M, Bhat G, Shahbaz Khan F, Felsberg M. 2017. Eco: Efficient convolution operators for tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 6638–6646.

Danelljan M, Häger G, Khan F, Felsberg M. 2014. Accurate scale estimation for robust visual tracking. In: British machine vision conference, Nottingham, September 1-5, 2014. Bmva Press.

Danelljan M, Hager G, Shahbaz Khan F, Felsberg M. 2015. Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE international conference on computer vision. p. 4310–4318.

Danelljan M, Robinson A, Shahbaz Khan F, Felsberg M. 2016. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14. Springer. p. 472–488.

Delforouzi A, Holighaus D, Grzegorzek M. 2020. Deep learning for object tracking in 360 degree videos. In: Progress in Computer Recognition Systems 11. Springer. p. 205–213.

Delforouzi A, Tabatabaei SAH, Shirahama K, Grzegorzek M. 2016. Unknown object tracking in 360-degree camera images. In: 2016 23rd International Conference on Pattern Recognition

(ICPR). IEEE. p. 1798–1803.
Friston S, Ritschel T, Steed A. 2019. Perceptual rasterization for head-mounted display image synthesis. ACM Trans Graph. 38(4):97–1.
Guo J, Huang L, Chien WC. 2022. Multi-viewport based 3d convolutional neural network for 360-degree video quality assessment. Multimedia Tools and Applications. 81(12):16813–16831.
Henriques JF, Caseiro R, Martins P, Batista J. 2014. High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence. 37(3):583–596.
Hong L, Yan S, Zhang R, Li W, Zhou X, Guo P, Jiang K, Chen Y, Li J, Chen Z, et al. 2024. Onetracker: Unifying visual object tracking with foundation models and efficient tuning. Available from: https://arxiv.org/abs/2403.09634.
Hu H, Ma B, Shen J, Sun H, Shao L, Porikli F. 2018. Robust object tracking using manifold regularized convolutional neural networks. IEEE Transactions on Multimedia. 21(2):510–521.
Huang H, Xu Y, Chen Y, Yeung SK. 2023. 360vot: A new benchmark dataset for omnidirectional visual object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. p. 20566–20576.
Huang Y, Li X, Zhou Z, Wang Y, He Z, Yang MH. 2024. Rtracker: Recoverable tracking via pn tree structured memory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. p. 19038–19047.
Kou S, Zhang FL, Lai YK, Dodgson NA. 2024. Neural panoramic representation for spatially and temporally consistent 360° video editing. In: 2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE. p. 200–209.
Li F, Tian C, Zuo W, Zhang L, Yang MH. 2018. Learning spatial-temporal regularized correlation filters for visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 4904–4913.
Li Y, Barnes C, Huang K, Zhang FL. 2022a. Deep 360 optical flow estimation based on multi-projection fusion. In: European Conference on Computer Vision. Springer. p. 336–352.
Li Y, Zhu J. 2015. A scale adaptive kernel correlation filter tracker with feature integration. In: Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II 13. Springer. p. 254–265.
Li YJ, Shi J, Zhang FL, Wang M. 2022b. Bullet comments for 360 video. In: 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE. p. 1–10.
Liao B, Wang C, Wang Y, Wang Y, Yin J. 2020. Pg-net: Pixel to global matching network for visual tracking. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. Springer. p. 429–444.
Liu KC, Shen YT, Chen LG. 2018. Simple online and realtime tracking with spherical panoramic camera. In: 2018 IEEE International Conference on Consumer Electronics (ICCE). IEEE. p. 1–6.
Mi TW, Yang MT. 2019. Comparison of tracking techniques on 360-degree videos. Applied Sciences. 9(16):3336.
Nasrabadi AT, Samiei A, Mahzari A, McMahan RP, Prakash R, Farias MC, Carvalho MM. 2019. A taxonomy and dataset for 360 videos. In: Proceedings of the 10th ACM Multimedia Systems Conference. p. 273–278.
Peng H, Zhang FL. 2024. Robust 360° visual tracking with dynamic gnomonic projection. In: 2024 39th International Conference on Image and Vision Computing New Zealand (IVCNZ). IEEE. p. 1–6.
Ravi N, Gabeur V, Hu YT, Hu R, Ryali C, Ma T, Khedr H, Rädle R, Rolland C, Gustafson L, et al. 2024. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:240800714.
Regensky A, Herglotz C, Kaup A. 2022. Motion-plane-adaptive inter prediction in 360-degree video coding. arXiv preprint arXiv:220203323.
Schroers C, Bazin JC, Sorkine-Hornung A. 2018. An omnistereoscopic video pipeline for capture and display of real-world vr. ACM Transactions on Graphics (TOG). 37(3):1–13.
Song H. 2014. Robust visual tracking via online informative feature selection. Electronics

Letters. 50(25):1931–1933.
Tursun OT, Arabadzhiyska-Koleva E, Wernikowski M, Mantiuk R, Seidel HP, Myszkowski K, Didyk P. 2019. Luminance-contrast-aware foveated rendering. ACM Transactions on Graphics (TOG). 38(4):1–14.
Wang L, Zhang L, Yi Z. 2017. Trajectory predictor by using recurrent neural networks in visual tracking. IEEE transactions on cybernetics. 47(10):3172–3183.
Wang M, Li YJ, Zhang WX, Richardt C, Hu SM. 2020. Transitioning360: Content-aware nfov virtual camera paths for 360 video playback. In: 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE. p. 185–194.
Wang Q, Yuan C, Wang J, Zeng W. 2018. Learning attentional recurrent neural network for visual tracking. IEEE Transactions on Multimedia. 21(4):930–942.
Wang Y, Zhang FL, Dodgson NA. 2024. Scantd: 360° scanpath prediction based on time-series diffusion. In: Proceedings of the 32nd ACM International Conference on Multimedia. p. 7764–7773.
Wax N. 1955. Signal-to-noise improvement and the statistics of track populations. Journal of Applied physics. 26(5):586–595.
Wu Y, Lim J, Yang MH. 2013. Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 2411–2418.
Xu L, Diao Z, Wei Y. 2022. Non-linear target trajectory prediction for robust visual tracking. Applied Intelligence:1–15.
Xu T, Feng ZH, Wu XJ, Kittler J. 2019a. Joint group feature selection and discriminative filter learning for robust visual object tracking. In: Proceedings of the IEEE/CVF international conference on computer vision. p. 7950–7960.
Xu T, Feng ZH, Wu XJ, Kittler J. 2019b. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. IEEE Transactions on Image Processing. 28(11):5596–5609.
Yang H, Shao L, Zheng F, Wang L, Song Z. 2011. Recent advances and trends in visual tracking: A review. Neurocomputing. 74(18):3823–3831.
Zhou Z, Chen J, Pei W, Mao K, Wang H, He Z. 2022. Global tracking via ensemble of local trackers. Available from: https://arxiv.org/abs/2203.16092.
Zhu H, Peng H, Xu G, Deng L, Cheng Y, Song A. 2021. Bilateral weighted regression ranking model with spatial-temporal correlation filter for visual tracking. IEEE Transactions on Multimedia. 24:2098–2111.