2023 IEEE International Symposium on Multimedia (ISM) | 979-8-3503-9576-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISM59092.2023.00010

360TripleView: 360-Degree Video View Management System Driven by Convergence Value of Viewing Preferences

Qian Zhou*, Mingyuan Wu*, Yinjie Zhang*, Michael Zink†, Ramesh Sitaraman‡ and Klara Nahrstedt*
*Department of Computer Science, University of Illinois Urbana-Champaign
†Department of Electrical and Computer Engineering, University of Massachusetts Amherst
‡College of Information and Computer Sciences, University of Massachusetts Amherst
Email: {qianz, mw34, yinjie2}@illinois.edu, mzink@cas.umass.edu, ramesh@cs.umass.edu, klara@illinois.edu

***Abstract*—360-degree video has become increasingly popular in content consumption. However, finding the viewing direction for important content within each frame poses a significant challenge. Existing approaches rely on either viewer input or algorithmic determination to select the viewing direction, but neither mode consistently outperforms the other in terms of content-importance. In this paper, we propose 360TripleView, the first view management system for 360-degree video that automatically infers and utilizes the better view mode for each frame, ultimately providing viewers with higher content-importance views. Through extensive experiments and a user study, we demonstrate that 360TripleView achieves over 90% accuracy in inferring the better mode and significantly enhances content-importance compared to existing methods.**

I. Introduction

Omnidirectional video (360° video) distinguishes itself from conventional 2D video by capturing a panoramic field of view. During playback, each 360° frame is projected onto a 2D view based on the viewer's viewing direction, resulting in an immersive and flexible viewing experience. 360° video aligns naturally with the growing demand for virtual and mixed reality applications, which industry giants such as Apple, Google, Microsoft, Meta, and others are heavily investing in, making it an integral part of our content consumption.

However, a crucial challenge lies in finding a 2D view from each 360° frame that provides important content to the viewer. Note that: 1) each 360° frame offers multiple potential viewing directions, and thus multiple potential views; 2) the same view can hold varying degrees of content-importance to different viewers, owing to their diverse viewing preferences. As a result, unless all possible views are examined, and the viewer's preference is taken into account, the selected view displayed to them may lack the desired importance.

Various approaches have been developed to identify important 2D views, with most falling into two categories. Firstly, many works [1], [2], [3], [4], [5] rely on manual viewer control to select views. In this approach, the viewer manually adjusts their viewing direction during playback, receiving the corresponding 2D view. By actively controlling the viewing direction, they can prioritize views that are highly important to them based on their individual viewing preference. However, due to the limited field of view of a human, the viewer can only observe one of the many possible views for each 360° frame. Consequently, while they can still discover important content within their field of view, they may overlook other views out of their sight that possess even higher content-importance.

Secondly, other works [6], [7], [8], [9], [10], [11] rely on algorithmic approaches, often built upon saliency detection [12]. In this scenario, a video server performs saliency detection on each 360° frame to identify the most salient 2D view. By systematically examining all possible views for each frame, it may discover views with higher content-importance compared to manually selected views. However, saliency detection algorithms do not consider the diverse viewing preferences of different viewers. As a result, the same view is recommended to all viewers, which can be of excellent importance to those who prefer to focus on salient objects but of low importance to viewers whose preferences do not align with saliency.

As a result, viewers face a dilemma: should they rely on their own instintcs to find views, which consistently offer high but not excellent content-importance, or watch algorithm-found views that fluctuate between excellent and low content-importance? Since neither view mode consistently outperforms the other in terms of content-importance, relying on a single mode throughout the video would result in limited content-importance. In this paper, we propose 360TripleView, the first intelligent 360° video view management system that addresses this dilemma by dynamically inferring the better view mode for each 360° frame. 360TripleView offers three view modes, each serving a specific purpose:

- **Manual.** Each viewer manually selects their views.
- **Auto$^{\text{optional}}$.** Algorithm-found views are provided, but viewers in Auto$^{\text{optional}}$ have the option to switch between Manual and Auto$^{\text{optional}}$.
- **Auto$^{\text{enforced}}$.** Algorithm-found views are provided, and no manual intervention is permitted in Auto$^{\text{enforced}}$.

The key to enhancing the overall content-importance for viewers in 360TripleView is its **View Mode Decision-Maker**, which automatically determines whether to utilize Auto$^{\text{enforced}}$ or Auto$^{\text{optional}}$/Manual for each

979-8-3503-9576-1/23/$31.00 ©2023 IEEE
DOI 10.1109/ISM59092.2023.00010

360° frame. It assesses whether the frame's algorithm-found views have higher content-importance than viewer-found views. If so, it employs AUTO$^{\text{ENFORCED}}$ to ensure that everyone observes the algorithm-found views. Otherwise, it utilizes AUTO$^{\text{OPTIONAL}}$/MANUAL, where the viewer can freely switch between algorithm-found views (AUTO$^{\text{OPTIONAL}}$) and their manually found views (MANUAL).

Inferring the better mode (with higher content-importance) between AUTO$^{\text{ENFORCED}}$ and AUTO$^{\text{OPTIONAL}}$/MANUAL is a problem that has never been studied. We tackle this challenge based on these insights: 1) Viewers' viewing preferences exhibit convergence in some 360° frames while divergence in others. We quantify such convergence using a novel metric referred to as the **Convergence Value of Viewing Preferences (CVVP)**. 2) We find that CVVP is instrumental in inferring the better mode between AUTO$^{\text{ENFORCED}}$ and AUTO$^{\text{OPTIONAL}}$/MANUAL. We develop a machine learning-based approach to automatically infer the CVVP of each 360° frame, based on which the view mode to use is determined.

Our contributions are as follows:

- In Section III, we introduce 360TripleView, the first view management system for 360° video that enhances overall content-importance for viewers by automatically inferring and utilizing the better mode between AUTO$^{\text{ENFORCED}}$ and AUTO$^{\text{OPTIONAL}}$/MANUAL for each 360° frame.
- In Section IV, we define the CVVP metric and propose a deep learning-based solution to estimate the CVVP for each frame automatically. During the offline stage, a few viewers' labeled viewing preferences on some videos are required to generate the ground truth CVVP for model training. When utilized, the model takes frames from new videos as input and returns the estimated CVVP, and no viewer needs to provide their viewing preference.
- Our experiments (Section V) and user study (Section VI) show that 360TripleView achieves an accuracy above 90% in inferring the better mode and delivers views of higher content-importance than existing approaches.

II. BACKGROUND AND RELATED WORK

360° Video Viewing. The nature of 360° video is depicted in Fig. 1, where it is an omnidirectional recording with a much wider field of view (**FoV**) than that of human eyes (horizontally $< 120°$). To make 360° video viewing intuitive, a sequence of viewing directions $\{(\psi_i, \theta_i)\}$ is provided, where the yaw angle $\psi_i \in [-180°, 180°]$, the pitch angle $\theta_i \in [-90°, 90°]$, and i represents the frame ID. These viewing directions allow each 360° frame to be projected onto a 2D view for viewing. In this paper, the terms "find a 2D view" and "find the viewing direction (ψ, θ)" are used interchangeably since determining the viewing direction of a 360° frame enables the identification of the corresponding 2D view.

Content-Importance. The importance of content within a 2D view varies according to the diverse viewing preferences of viewers. In this paper, we focus on developing a view management system that automatically infers the better view mode to utilize, making viewers obtain views with higher content-importance overall.

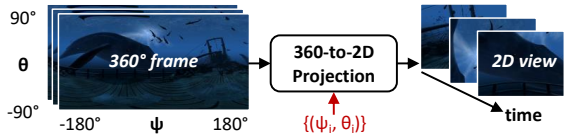


Fig. 1: Projection of 360° frames onto 2D views based on a sequence of viewing directions.

Existing view modes can be categorized as follows:

A. MANUAL *Mode*

In MANUAL mode, the viewer has complete control over their viewing direction (ψ, θ). Using a client-device (e.g., a mouse or headset), the viewer manually adjusts their viewing direction throughout the video. The client-device continuously sends the updated (ψ, θ) to the video server, which keeps returning the corresponding 2D view to the viewer.

Pros & Cons. Many existing works [1], [2], [3], [4], [5] consider MANUAL to be satisfactory for viewers. This is because manual control allows viewers to obtain views that align with their own preferences and are therefore important to them. However, due to the viewer's limited FoV, they can only see a small portion of the entire 360° frame at a given time. Consequently, they may perceive the view they are watching as the most important, while a more significant view exists outside of their sight, which they would have turned to if they had been aware of it. As a result, MANUAL *consistently delivers views of high content-importance to the viewer but not of excellent importance*.

To address the drawback of the limited FoV, some works [13], [14] introduce graphical indicators within the viewer's FoV to indicate targets outside their sight. However, these solutions require the viewer to manually track multiple view candidates, which can be distracting or overwhelming.

B. AUTO$^{\text{ENFORCED}}$ *Mode*

On the other hand, AUTO$^{\text{ENFORCED}}$ mode offers no control to the viewer. In this mode, a server performs saliency detection [15], [12] to determine the viewing direction (ψ, θ) that results in the most salient 2D view, which is then delivered to the viewer. In AUTO$^{\text{ENFORCED}}$, the viewer can only watch the auto-generated 2D video and cannot switch to another mode.

Pano2Vid [6] converts a 360° video into a 2D video that resembles those captured by human videographers. Deep360Pilot [7] utilizes supervised learning, where the authors manually label the most salient object frame by frame and train an RNN to recommend the corresponding (ψ, θ). Wang et al. [8] employ reinforcement learning using ground truth data from the Pano2Vid and Deep360Pilot datasets, combined with saliency detection. Lai et al. [9] incorporate saliency detection and semantic segmentation. They also propose saliency-aware temporal summarization, which increases the playback speed of frames with lower saliency scores.

Pros & Cons. Since the algorithm considers the entire 360° frame and explores all possible 2D views, it may discover a view of higher content-importance compared to what a viewer might find in MANUAL mode. However, due to the diverse viewing preferences among viewers, an algorithm-found view may be of excellent importance to some viewers but have little or no importance to others. Personalization through machine learning, which recommends different views to different viewers based on their preferences, is not feasible at this stage since obtaining every individual's 360° video viewing preference is impractical. Therefore, in this paper, we assume that AUTO$^{\text{ENFORCED}}$ recommends the same 2D view to all viewers who watch the same 360° frame, without personalization.

As a result, viewers using AUTO$^{\text{ENFORCED}}$ mode risk receiving a view that may be more or less important compared to what they would obtain in MANUAL mode. In other words, *neither* AUTO$^{\text{ENFORCED}}$ *nor* MANUAL *is consistently superior to the other in terms of content-importance*.

C. AUTO$^{\text{OPTIONAL}}$/MANUAL *Mode*

In AUTO$^{\text{OPTIONAL}}$/MANUAL mode [16], [17], the viewer can manually switch between algorithm-found views (AUTO$^{\text{OPTIONAL}}$) and viewer-found views (MANUAL) whenever desired. The algorithm-found views in AUTO$^{\text{OPTIONAL}}$ are the same as those in AUTO$^{\text{ENFORCED}}$. The key distinction is that the viewer can interrupt AUTO$^{\text{OPTIONAL}}$ and switch to MANUAL, but such interruptions are not possible in AUTO$^{\text{ENFORCED}}$.

These works aim to enhance content-importance by providing viewers with two mode options, allowing them to choose the better one. However, *they overlook the fact that humans are incapable of accurately and instantly determining the better mode for each frame without being distracted from enjoying the video content*.

III. OVERVIEW OF 360TRIPLEVIEW

Our 360TripleView has the following three view modes, and utilizes one mode at a time.

- **MANUAL.** Each viewer manually selects their views.
- **AUTO$^{\text{OPTIONAL}}$.** Algorithm-found views are provided, but viewers in AUTO$^{\text{OPTIONAL}}$ have the option to switch between MANUAL and AUTO$^{\text{OPTIONAL}}$.
- **AUTO$^{\text{ENFORCED}}$.** Algorithm-found views are provided, and no manual intervention is permitted in AUTO$^{\text{ENFORCED}}$.

As illustrated in Fig. 2, 360TripleView involves the server processing and transmitting video content, while viewers access the content through their client-devices (e.g., headsets). The **View Mode Decision-Maker** on the server receives two inputs: ❶ a 360° frame from the Video Database, and ❷ the viewer's request to change their view mode. The decision-maker determines ❸ the view mode to use for each frame and sends the decision (denoted as $mode_{use} \in \{$MANUAL, AUTO$^{\text{OPTIONAL}}$, AUTO$^{\text{ENFORCED}}\}$) to the viewer, the saliency detection unit, and the 360-to-2D projection unit. If $mode_{use}$ is MANUAL, the viewer continues to provide ❹ their manually controlled viewing direction (ψ, θ) to the projection unit; if $mode_{use}$ is AUTO$^{\text{OPTIONAL}}$ or AUTO$^{\text{ENFORCED}}$, the saliency detection unit processes the video frame and ❺ automatically recommends a (ψ, θ) to the projection unit. It is important to note that 360TripleView can utilize any existing saliency detection approach (e.g., we use ATSal [12], but other solutions also work). Finally, the projection unit receives the video frame and the (ψ, θ) (either ❹ or ❺, depending on $mode_{use}$), and delivers ❻ the corresponding 2D view to the viewer.

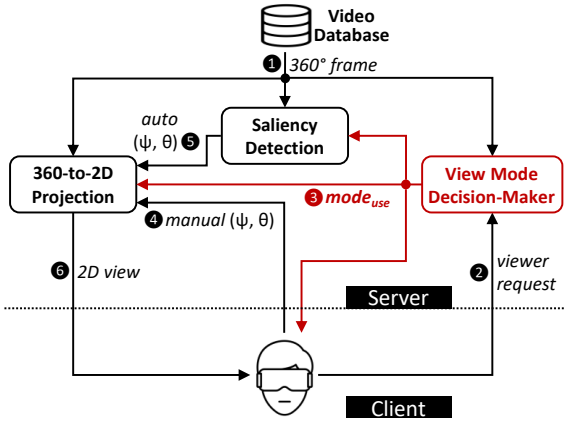


Fig. 2: 360TripleView system architecture.

A. View Mode Decision-Maker

The cornerstone of 360TripleView (Fig. 2) is its View Mode Decision-Maker (Fig. 3). This component determines, for each 360° frame, which of the three view modes (MANUAL, AUTO$^{\text{OPTIONAL}}$, AUTO$^{\text{ENFORCED}}$) to utilize (i.e., be $mode_{use}$).

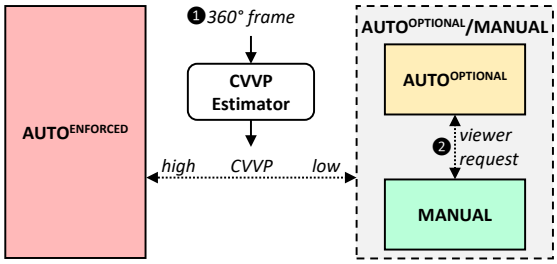


Fig. 3: 360TripleView View Mode Decision-Maker..

As depicted in Fig. 3, the View Mode Decision-Maker operates as a state machine with three states: MANUAL, AUTO$^{\text{OPTIONAL}}$, AUTO$^{\text{ENFORCED}}$. The **CVVP Estimator** is responsible for determining whether to employ AUTO$^{\text{ENFORCED}}$ or AUTO$^{\text{OPTIONAL}}$/MANUAL. It takes ❶ a 360° frame as input and estimates the **Convergence Value of Viewer Preferences (CVVP)**. The CVVP is a metric that indicates the better mode, either AUTO$^{\text{ENFORCED}}$ or AUTO$^{\text{OPTIONAL}}$/MANUAL, in terms of higher content-importance. We will provide further details on this metric in Section IV. If the CVVP exceeds a threshold, it suggests that algorithm-found views may offer higher overall content-importance compared to viewer-found

views. Therefore, in such cases, $mode_{use}$ will be set to AUTO$^{\text{ENFORCED}}$ to ensure that algorithm-found views are not missed by any viewer. Conversely, a low CVVP implies that viewer-found views possess higher content-importance. Consequently, $mode_{use}$ will be set to AUTO$^{\text{OPTIONAL}}$/MANUAL, allowing each viewer to manually send ❷ a viewer request (e.g., through mouse clicks) and switch the $mode_{use}$ between MANUAL and AUTO$^{\text{OPTIONAL}}$ (without affecting other viewers' $mode_{use}$). It suggests viewers select MANUAL to watch their personally found views, but allows them the option to watch algorithm-found views (AUTO$^{\text{OPTIONAL}}$) if desired, especially when they feel fatigued after prolonged use of MANUAL.

IV. VIEW MODE DECISION-MAKING DRIVEN BY CVVP

360TripleView's View Mode Decision-Maker automatically infers whether AUTO$^{\text{ENFORCED}}$ or AUTO$^{\text{OPTIONAL}}$/MANUAL will yield higher content-importance, based on a novel metric called the Convergence Value of Viewer Preferences (CVVP). In this section, we elaborate on CVVP and its use in inferring the better mode, and present a deep learning solution to automatically estimate the CVVP for each 360° frame.

A. Definition of CVVP

Through experiments, we have made a key observation: when multiple viewers are presented with a complete view of a 360° frame, and each viewer is asked to identify the viewing direction that holds their highest content-importance. Their preferences—indicated by their labeled directions—diverge in some frames while converge in others. Figure 4 illustrates the variation of ψ (yaw) labeled by different viewers over time in a video from the Pano2Vid [6] dataset (`www.youtube.com/watch?v=i9SiIyCyRM0`): their preferences diverge from second 57 to 80 and converge well at other times. Divergence often occurs when a frame contains *zero or multiple* significant regions, leading to different choices based on individual preferences. Conversely, convergence occurs when a frame contains *one* dominant important region which is favored by most viewers. Consequently, viewers' viewing preferences exhibit a dynamic degree of convergence that varies across video frames.

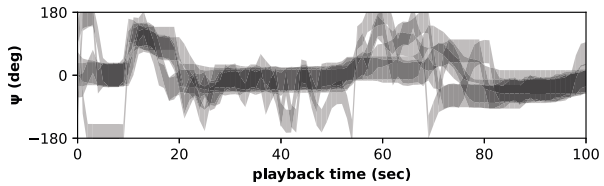


Fig. 4: Variation of viewers' labeled ψ (yaw), showing convergence and divergence. Variation also exists for θ (pitch).

To quantify the degree of convergence for each 360° frame, we introduce a novel metric—Convergence Value of Viewer Preferences (CVVP). Computing the **ground truth CVVP** requires the most important labeled viewing directions from N viewers, denoted as (ψ_j, θ_j), where the viewer ID is represented by $j = 1, 2, \ldots, N$. The **content-importance** of a viewing direction (ψ, θ) to viewer j is defined as follows:

$$importance_j(\psi,\theta) = \begin{cases} 1 & \text{if gcd}((\psi,\theta),(\psi_j,\theta_j)) < TH_{dist} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where gcd() is the great-circle distance between two viewing directions (each direction corresponds to a point on the unit sphere), and TH_{dist} is the distance threshold. Specifically, (ψ, θ) is considered important to viewer j if it is close enough to the viewer's labeled direction. Given that the human field of view is $< 120°$, we consider two directions to be close if they are less than 30° apart. Thus, we set TH_{dist} to 30°.

The **overall content-importance** of (ψ, θ) is defined as the average of the content-importance values across all N viewers:

$$importance(\psi,\theta) = \frac{1}{N}\sum_{j=1}^{N} importance_j(\psi,\theta) \tag{2}$$

Finally, the **CVVP** of the 360° frame is defined as the maximum $importance(\psi, \theta)$ among all $\psi \in [-180°, 180°]$, $\theta \in [-90°, 90°]$:

$$\text{CVVP} = \max_{\psi,\theta} importance(\psi,\theta) \tag{3}$$

Figure 5 presents two examples of frames, their labels, and ground truth CVVP values.

(a) CVVP $= \frac{2}{6} \approx 0.33$ (b) CVVP $= \frac{6}{6} = 1$

Fig. 5: Examples of ground truth CVVP. Each circle represents a viewer-labeled most important viewing direction. Directions close to each other are color-coded identically.

B. Using CVVP to Infer the Better Mode

The View Mode Decision-Maker utilizes CVVP to infer the better mode (with higher overall content-importance), between AUTO$^{\text{ENFORCED}}$ and AUTO$^{\text{OPTIONAL}}$/MANUAL. The inferred better mode is then selected as the mode to use ($mode_{use}$):

$$mode_{use} = \begin{cases} \text{AUTO}^{\text{OPTIONAL}}\text{/MANUAL} & \text{if CVVP} < TH_{CVVP} \\ \text{AUTO}^{\text{ENFORCED}} & \text{otherwise} \end{cases} \tag{4}$$

where TH_{CVVP} represents a configurable CVVP threshold.

Why is CVVP an effective indicator for inferring the better mode? Note that CVVP $\in (0, 1]$ increases with the convergence degree of viewing preferences:

1) In the case of the lowest convergence, where (ψ_j, θ_j) are scattered, any (ψ, θ) is close to at most one (ψ_j, θ_j), i.e., $\forall(\psi,\theta),\ importance(\psi,\theta) \leq 1$, thus CVVP $= \frac{1}{N}$.

2) In the case of the highest convergence, where all N viewers' (ψ_j, θ_j) are closely clustered, there exists a (ψ, θ) that is close to all of them, i.e., $\exists(\psi, \theta),\ importance(\psi, \theta) = N$, thus CVVP $= 1$.
3) In general, if a 360° frame has a CVVP of $\eta\%$, it implies that at most $\eta\%$ of the viewers will have their preferred view when one (ψ, θ) is viewed by all viewers. Notably, AUTO$^{\text{ENFORCED}}$ recommends one (ψ, θ) to all viewers without personalization (Section II). Thus, CVVP serves as an upper bound for the actual overall content-importance achieved by AUTO$^{\text{ENFORCED}}$.

Based on these observations, we make the inferences below:

- If CVVP $< TH_{CVVP}$ (e.g., 60%), the actual overall content-importance achieved by AUTO$^{\text{ENFORCED}}$ must be $< TH_{CVVP}$. Thus, AUTO$^{\text{OPTIONAL}}$/MANUAL is inferred as the better mode and becomes $mode_{use}$, allowing viewers to use MANUAL.
- If CVVP $\geq TH_{CVVP}$, the actual overall content-importance achieved by AUTO$^{\text{ENFORCED}}$ can be $\geq TH_{CVVP}$. Thus, AUTO$^{\text{ENFORCED}}$ is inferred as the better mode and becomes $mode_{use}$, ensuring that algorithm-found views are watched.

C. Automatic CVVP Estimator

To compute the ground truth CVVP (Section IV-A) of a 360° frame, we need to know the important viewing directions labeled by multiple viewers for that frame. However, obtaining this information from viewers during the use of 360TripleView is impractical. Therefore, we introduce a deep learning solution: in the offline stage, we request a few viewers to label some frames of some videos, enabling us to compute the ground truth CVVP for model training; when utilized, the model processes frames from new videos (not used in training) and provides the estimated CVVP without requiring any viewers to provide their viewing preferences.

1) Deep Learning-Based Regression: We have developed a deep learning-based regression model that takes a 360° frame (represented as I) as input and predicts its CVVP $\in \mathbb{R}$ and $\in (0, 1]$. The model leverages ResNet101 for image feature extraction. Since ResNet is pretrained on 2D images and does not handle equirectangular 360° frames, which are significantly distorted in the polar regions, we first convert I to a cubemap comprising six 2D views denoted as $\{I^x\}$, where $x =$ front, back, left, right, up, down. Each view is passed through the feature extractor to obtain its visual features $V_{I^x} \in \mathbb{R}^{2048}$. These features are concatenated to $V_I \in \mathbb{R}^{12288}$.

V_I is fed through two fully connected layers with output sizes of 2048 and 1 (representing the predicted CVVP), respectively. ReLU activation is applied to each layer. Mean Absolute Error (L1) is used as the loss function instead of Mean Squared Error (L2) because MAE is more robust against outliers. Fig. 6 displays the predicted CVVP and the ground truth for the video in Fig. 4. It is seen that the predicted and ground truth values are relatively low from second 57 to 80, consistent with the divergence depicted in Fig. 4.

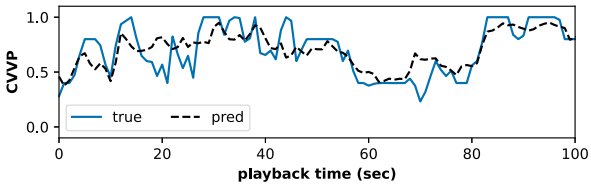


Fig. 6: CVVP predicted by the neural network.

2) Binarization and Stabilization: The regression model outputs a sequence of CVVP values $\{CVVP_i\}$, where i represents the frame ID. Note that if $mode_{use}$ changes with CVVP frame by frame, viewers will be disturbed. To stabilize it, we compute the average CVVP per second, getting $\{CVVP_t\}$ where $t = 1, 2, \ldots, T$ second, $CVVP_t \in \mathbb{R}$ and $\in (0, 1]$. However, we notice that $CVVP_t$ still fluctuates often. If we simply binarize the $\{CVVP_t\}$ in Fig. 6 using threshold TH_{CVVP} (e.g., 0.6, Section IV-B), making a CVVP above TH_{CVVP} be 1 (indicating the use of AUTO$^{\text{ENFORCED}}$) and otherwise be 0 (indicating the use of AUTO$^{\text{OPTIONAL}}$/MANUAL), we will get the sequence shown in Fig. 7, with overly short view modes and frequent view mode switching.

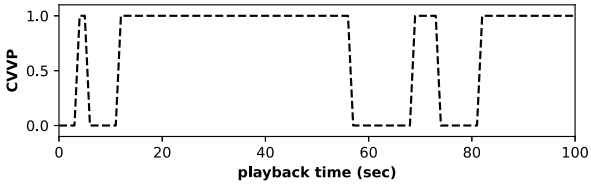


Fig. 7: Predicted CVVP sequence without stabilization.

To address this issue, we devise an advanced stabilization module, which takes $\{CVVP_t\}$ as input, along with two parameters set by the system administrator: 1) TH_{CVVP}, the threshold above which a CVVP is considered high enough to use AUTO$^{\text{ENFORCED}}$; 2) t_{min}, the minimum duration (e.g., 20 seconds) for which the view mode must remain unchanged before it can change again, ensuring stability.

First, we normalize $\{CVVP_t\}$ to $\{CVVP'_t\}$ such that $CVVP_t = TH_{CVVP}$ is mapped to 0.5. Then, we binarize $\{CVVP'_t\}$ to $\{\overline{CVVP}_t\}$, where $\overline{CVVP}_t \in \{0, 1\}$. The objective is to minimize the difference between $\{\overline{CVVP}_t\}$ and $\{CVVP'_t\}$ while ensuring that each time $\overline{CVVP}_t$ changes (from 0 to 1 or 1 to 0), the new value persists for at least t_{min} seconds. We can formulate the problem as follows:

$$\begin{aligned} \underset{\{\overline{CVVP}_t\}_{v,[t_1,t_2,\ldots,t_m]}}{\arg\min} \quad & \text{MSE}(\{\overline{CVVP}_t\}_{v,[t_1,t_2,\ldots,t_m]}, \{CVVP'_t\}) \\ \text{s.t.} \quad & v \in \{0, 1\} \\ & \sum_{i=1}^{m} t_i = T \quad 1 \leq m \leq \left\lfloor \frac{T}{t_{min}} \right\rfloor, t_i \geq t_{min} \end{aligned} \tag{5}$$

where v represents the initial value of $\{\overline{CVVP}_t\}$ (0 or 1), and $[t_1, t_2, \ldots, t_m]$ indicates that $\{\overline{CVVP}_t\}$ consists of m disjoint subsequences, with t_i being the length of the ith subsequence. Thus, $\{\overline{CVVP}_t\}_{v,[t_1,t_2,\ldots,t_m]}$ represents the sequence that starts with the value v, lasts for t_1 seconds, toggles the value (0 to 1 or 1 to 0), lasts for t_2 seconds, and so on. The second constraint ensures that all t_i add up to T (the total length of the sequence) and that each subsequence is at least t_{min} seconds long. The optimal solution, which minimizes the mean squared error (MSE) to $\{CVVP'_t\}$, is the resulting sequence $\{\overline{CVVP}_t\}$. With $t_{min} = 20$ seconds, the stabilized result of the sequence in Fig. 7 is illustrated in Fig. 8.

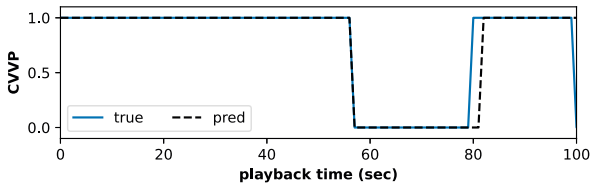


Fig. 8: CVVP sequence with stabilization.

V. Experiments

A. Implementation and Experimental Settings

1) Dataset: To train and test the deep learning model, we acquire 360° videos with ground truth CVVP. We utilize the Pano2Vid dataset [6], which has videos of diverse content, such as tours, sports, and parades. Each frame of each video in Pano2Vid has most important (ψ, θ) labeled by different participants (see examples in Fig. 5). We convert these labels to ground truth CVVP per frame, following the definition of CVVP (Section IV-A). It is worth noting that Pano2Vid is the only dataset that meets our criterion for ground truth CVVP generation, because *each participant is given the entire view of a 360° frame and asked to check every viewing direction and label the highest content-importance view according to their viewing preference, frame by frame.* More recent 360° video datasets [18], [19] are collected by having participants watch videos with headsets in MANUAL mode while recording their viewing directions in real time. However, MANUAL cannot guarantee that the most important (ψ, θ) is found, because of its limited field of view. Therefore, these datasets are unsuitable for generating ground truth CVVP.

2) Deep Learning: For feature extraction in our CVVP regression model, we use ResNet101. We have experimented with other models, such as VGG19 and Inception-v3, but found no significant impact on the accuracy of CVVP estimation. We conduct three validation schemes:

- **No Tuning (leave-one-out):** Each video is tested using the model trained on the other videos. This is a commonly used cross-validation scheme.
- **1-sec Tuning:** For each video, we randomly select 1 second (30 frames) of its content and use the corresponding ground truth CVVP to fine-tune the model trained on the other videos, and then test the video. *Note that tuning is not obligatory; it can be selected when available to improve CVVP prediction accuracy.*
- **3-sec Tuning:** This scheme is similar to 1-sec Tuning, but we use 3-second ground truth CVVP for fine-tuning.

3) Evaluation Metrics:

- **Error of Estimated CVVP:** This metric represents the difference between the predicted CVVP per frame and the ground truth CVVP.
- **Accuracy of Better Mode Inference:** A true positive (TP) occurs when both predicted and ground truth CVVPs are 1 (indicating the use of AUTO$^{\text{ENFORCED}}$). A true negative (TN) occurs when both are 0 (indicating the use of AUTO$^{\text{OPTIONAL}}$/MANUAL). $accuracy = \frac{TP+TN}{\text{total}}$.
- **Overall Content-Importance:** It measures the actual overall content-importance (Equation 2 in Section IV-A).

4) Baselines: 360TripleView automatically determines $mode_{use}$ based on CVVP. We compare it with two baseline methods that determine $mode_{use}$.

- **AUTO$^{\text{ENFORCED}}$ ONLY:** $mode_{use}$ is AUTO$^{\text{ENFORCED}}$ from beginning to end.
- **AUTO$^{\text{OPTIONAL}}$/MANUAL ONLY:** $mode_{use}$ manually switches between AUTO$^{\text{OPTIONAL}}$ and MANUAL.

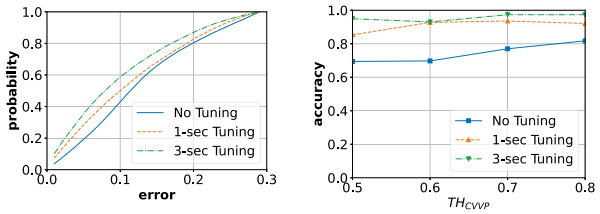
Note that 360TripleView innovates in $mode_{use}$ determination (i.e., deciding which mode to use), not in saliency detection. When $mode_{use}$ becomes AUTO$^{\text{ENFORCED}}$ or AUTO$^{\text{OPTIONAL}}$, the saliency detection unit (Fig. 2) executes an existing saliency detection algorithm, and its performance impacts the resulting overall content-importance. We test the following saliency detection approaches:

- **CubePad:** CubePadding [15] is a seminal and well-known saliency detection approach for 360° video.
- **ATSal:** ATSal [12] is one of the most recent state-of-the-art saliency detection methods for 360° video.
- **Pano2Vid:** The (ψ, θ) in Pano2Vid [6] are manually labeled, not algorithmically derived like CubePad or ATSal. We include it here because it represents the "ceiling" of (ψ, θ) recommendation, which may be approached by future algorithms.

B. Error of Estimated CVVP

Fig. 9a shows the cumulative distribution function (CDF) of the error of predicted CVVP for each validation scheme. All of them have an error within 0.15 for about 70% of the time, and an error within 0.25 for more than 90% of the time.

Unsurprisingly, the error decreases as more data are used for fine-tuning. But even without any tuning, it still achieves a mean error of 0.19. For 1-sec Tuning and 3-sec Tuning, the mean errors are 0.14 and 0.12, respectively. Considering that the range of CVVP is $(0, 1]$, these errors may not be very small. However, it is important to note that CVVP will be binarized and stabilized before being used to control view mode switching. For example, a CVVP of 0.7 (with its ground truth being 0.9) will be binarized to 1—the same as the binarized ground truth if the threshold TH_{CVVP} (Section IV-B) is 0.6.



(a) Error of estimated CVVP. (b) Accuracy of better mode inference.

Fig. 9: Performance of CVVP estimation and mode inference.

Thus, errors of this level do not prevent 360TripleView from overall accurately inferring the better mode (Section V-C).

C. Accuracy of Better Mode Inference

Fig. 9b shows the accuracy in inferring the better mode between AUTO$^{\text{ENFORCED}}$ and AUTO$^{\text{OPTIONAL}}$/MANUAL. Even without any tuning, the accuracy is still around 80%. When 1-sec Tuning is used, the accuracy is raised to above **90%** most of the time. The mean accuracy of each scheme is 74%, 91% and 96%, respectively. The threshold TH_{CVVP} varying from 0.5 to 0.8 has no significant impact on the accuracy.

D. Overall Content-Importance

We compare the overall content-importance when using AUTO$^{\text{ENFORCED}}$ ONLY, AUTO$^{\text{OPTIONAL}}$/MANUAL ONLY, and our 360TripleView. Note that for some videos, 360TripleView infers that AUTO$^{\text{ENFORCED}}$ is the better mode throughout the video, resulting in the same content-importance as the first baseline (AUTO$^{\text{ENFORCED}}$ ONLY). To focus on the performance difference, we exclude those videos and present the average content-importance of the remaining videos whose content-importance varies with $mode_{use}$ determination strategies.

Impact of $mode_{use}$ Determination Strategies: The impact of $mode_{use}$ determination strategies when $TH_{CVVP} = 0.6$ is shown in Table II. It demonstrates that our 360TripleView achieves higher overall content-importance than the other $mode_{use}$ determination strategies (AUTO$^{\text{ENFORCED}}$ ONLY, AUTO$^{\text{OPTIONAL}}$/MANUAL ONLY) in almost all cases, when the saliency detection strategy (CubePad, ATSal, Pano2Vid) and the tuning time are held constant. Similar results are observed when $TH_{CVVP} = 0.5$ (Table I) and 0.7 (Table III).

Impact of Saliency Detection Strategies: 360TripleView focuses on $mode_{use}$ determination and does not propose a new saliency detection solution. When $mode_{use}$ becomes AUTO$^{\text{ENFORCED}}$ or AUTO$^{\text{OPTIONAL}}$, an existing saliency detection approach is employed (Fig. 2). Comparing CubePad with ATSal, we observe that the latter generally achieves higher content-importance. This is because both approaches infer content-importance based on saliency detection, but ATSal, being a more recent work, combines global and local visual features to predict saliency more accurately compared to previous methods. However, when comparing ATSal with Pano2Vid, a significant difference is still evident. This suggests

TABLE I: Overall content-importance ($TH_{CVVP} = 0.5$). The best performance value is marked in **bold**.

		CubePad	ATSal	Pano2Vid
No Tuning	AUTO$^{\text{ENFORCED}}$	0.150	0.208	0.582
	AUTO$^{\text{OPT}}$/MAN	0.137	**0.215**	0.599
	360TripleView	**0.199**	0.211	**0.636**
1-sec Tuning	AUTO$^{\text{ENFORCED}}$	0.123	0.145	0.507
	AUTO$^{\text{OPT}}$/MAN	0.097	**0.160**	0.482
	360TripleView	**0.188**	0.155	**0.613**
3-sec Tuning	AUTO$^{\text{ENFORCED}}$	0.094	0.151	0.545
	AUTO$^{\text{OPT}}$/MAN	0.068	0.130	0.512
	360TripleView	**0.154**	**0.170**	**0.647**

TABLE II: Overall content-importance ($TH_{CVVP} = 0.6$).

		CubePad	ATSal	Pano2Vid
No Tuning	AUTO$^{\text{ENFORCED}}$	**0.293**	0.411	0.782
	AUTO$^{\text{OPT}}$/MAN	0.230	0.387	0.727
	360TripleView	0.115	**0.498**	**0.798**
1-sec Tuning	AUTO$^{\text{ENFORCED}}$	0.089	0.187	0.594
	AUTO$^{\text{OPT}}$/MAN	0.086	0.157	0.574
	360TripleView	**0.119**	**0.230**	**0.723**
3-sec Tuning	AUTO$^{\text{ENFORCED}}$	0.112	0.218	0.603
	AUTO$^{\text{OPT}}$/MAN	0.098	0.224	0.580
	360TripleView	**0.189**	**0.239**	**0.720**

TABLE III: Overall content-importance ($TH_{CVVP} = 0.7$).

		CubePad	ATSal	Pano2Vid
No Tuning	AUTO$^{\text{ENFORCED}}$	**0.368**	0.616	0.855
	AUTO$^{\text{OPT}}$/MAN	0.133	**0.809**	0.877
	360TripleView	0.056	0.716	**0.962**
1-sec Tuning	AUTO$^{\text{ENFORCED}}$	0.097	0.268	0.652
	AUTO$^{\text{OPT}}$/MAN	0.092	0.246	0.640
	360TripleView	**0.144**	**0.299**	**0.773**
3-sec Tuning	AUTO$^{\text{ENFORCED}}$	0.141	0.296	0.657
	AUTO$^{\text{OPT}}$/MAN	0.081	0.311	0.660
	360TripleView	**0.242**	**0.312**	**0.781**

that the algorithm-generated (ψ, θ) values are currently quite distinct from the human-labeled ones, indicating room for improvement. While saliency detection falls outside the scope of this paper, we consider it as a potential area for future work.

Impact of Tuning: We observe that when the $mode_{use}$ determination and saliency detection strategies are held constant, the performance does not always increase with tuning time. This is because the no-tuning model wrongly uses AUTO$^{\text{ENFORCED}}$ consistently on many videos. As previously mentioned, if 360TripleView consistently uses AUTO$^{\text{ENFORCED}}$ on a video, it is essentially equivalent to AUTO$^{\text{ENFORCED}}$ ONLY, so we exclude the video. Consequently, the no-tuning model excludes more videos, which may make the average content-importance of the remaining videos higher.

VI. USER STUDY

A. User Study Settings

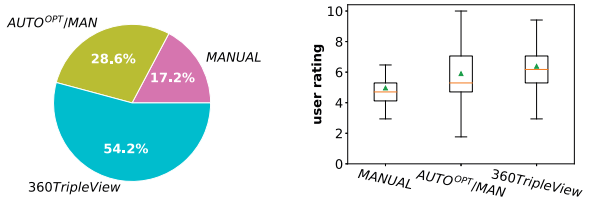
We design and implement an online platform, utilizing it to conduct a user study. A total of 25 participants (21

males and 4 females) are recruited. The dataset used for evaluation is the Pano2Vid dataset, and we select 6 videos that represent diverse content, including tours (hiking/driving), sports (outdoor/indoor), and parades (daytime/nighttime).

User Ratings: Each participant watches each video under three view modes sequentially: 1) MANUAL ONLY, 2) AUTO$^{\text{OPTIONAL}}$/MANUAL ONLY, and 3) 360TripleView. Participants are asked to rate the content-importance for each video and each mode on a scale from 0 (worst) to 10 (best).

B. User Study Results

Favorite Mode: If a view mode receives a higher rating than the other two modes from a participant, it is regarded as the participant's favorite mode. Fig. 10a shows the statistics: MANUAL ONLY is favored 17.2% of the time, AUTO$^{\text{OPTIONAL}}$/MANUAL ONLY is favored 28.6% of the time, while 360TripleView emerges as the clear winner, being the favorite mode 54.2% of the time.



(a) Favorite mode.

(b) User rating distribution (green: mean, orange: median).

Fig. 10: User study results.

User Ratings: The distribution of user ratings for content-importance in each view mode is presented in Fig. 10b. The mean ratings for MANUAL ONLY, AUTO$^{\text{OPTIONAL}}$/MANUAL ONLY and 360TripleView are 4.96, 5.90 and 6.38, respectively. The corresponding median ratings are 4.71, 5.29 and 6.18. It is evident that 360TripleView receives the highest ratings in both measures.

VII. CONCLUSION

In this paper, we have presented the design, implementation, and evaluation of 360TripleView, a groundbreaking view management system for 360° video viewing. It offers three view modes and automatically infers the better mode between AUTO$^{\text{ENFORCED}}$ and AUTO$^{\text{OPTIONAL}}$/MANUAL to enhance viewers' overall content-importance. Our evaluation results demonstrate that 360TripleView achieves an accuracy above 90% in inferring the better mode and results in significantly higher content-importance compared to existing approaches.

ACKNOWLEDGMENT

This work was funded by the National Science Foundation under grant contracts NSF 1835834, NSF 1900875, NSF 1901137 and NSF 2106592. Any results and opinions are our own and do not represent views of National Science Foundation.

REFERENCES

[1] A. Samiei and R. Prakash, "Improving 360-degree video field-of-view prediction and edge caching," in *2021 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2021, pp. 9–16.

[2] F.-Y. Chao, C. Ozcinar, and A. Smolic, "Transformer-based long-term viewport prediction in 360° video: Scanpath is all you need." in *MMSP*, 2021, pp. 1–6.

[3] S. Vats, J. Park, K. Nahrstedt, M. Zink, R. Sitaraman, and H. Hellwagner, "Semantic-aware view prediction for 360-degree videos at the 5g edge," in *2022 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2022, pp. 121–128.

[4] Q. Guimard, L. Sassatelli, F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, "Deep variational learning for multiple trajectory prediction of 360° head movements," in *Proceedings of the 13th ACM Multimedia Systems Conference*, 2022, pp. 12–26.

[5] Q. Zhou, Z. Yang, H. Guo, B. Tian, and K. Nahrstedt, "360broadview: Viewer management for viewport prediction in 360-degree video live broadcast," in *Proceedings of the 4th ACM International Conference on Multimedia in Asia*, 2022, pp. 1–7.

[6] Y.-C. Su, D. Jayaraman, and K. Grauman, "Pano2vid: Automatic cinematography for watching 360 videos," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 154–171.

[7] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun, "Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1396–1405.

[8] J. Wang, M. Xu, L. Jiang, and Y. Song, "Attention-based deep reinforcement learning for virtual cinematography of 360 videos," *IEEE Transactions on Multimedia*, vol. 23, pp. 3227–3238, 2020.

[9] W.-S. Lai, Y. Huang, N. Joshi, C. Buehler, M.-H. Yang, and S. B. Kang, "Semantic-driven generation of hyperlapse from 360 degree video," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 9, pp. 2610–2621, 2017.

[10] Y. Yu, S. Lee, J. Na, J. Kang, and G. Kim, "A deep ranking model for spatio-temporal highlight detection from a 360○ video," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[11] S. Lee, J. Sung, Y. Yu, and G. Kim, "A memory network approach for story-based temporal summarization of 360 videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1410–1419.

[12] Y. Dahou, M. Tliba, K. McGuinness, and N. O'Connor, "Atsal: An attention based architecture for saliency prediction in 360 videos," in *International Conference on Pattern Recognition*. Springer, 2021, pp. 305–320.

[13] Y.-C. Lin, Y.-J. Chang, H.-N. Hu, H.-T. Cheng, C.-W. Huang, and M. Sun, "Tell me where to look: Investigating ways for assisting focus in 360 video," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 2535–2545.

[14] Y.-T. Lin, Y.-C. Liao, S.-Y. Teng, Y.-J. Chung, L. Chan, and B.-Y. Chen, "Outside-in: Visualizing out-of-sight regions-of-interest in a 360 video using spatial picture-in-picture previews," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017, pp. 255–265.

[15] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun, "Cube padding for weakly-supervised saliency prediction in 360 videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1420–1429.

[16] S. Cha, J. Lee, S. Jeong, Y. Kim, and J. Noh, "Enhanced interactive 360 viewing via automatic guidance," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 5, pp. 1–15, 2020.

[17] M. Wang, Y.-J. Li, W.-X. Zhang, C. Richardt, and S.-M. Hu, "Transitioning360: Content-aware nfov virtual camera paths for 360° video playback," in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2020, pp. 185–194.

[18] Q. Guimard, F. Robert, C. Bauce, A. Ducreux, L. Sassatelli, H.-Y. Wu, M. Winckler, and A. Gros, "Pem360: A dataset of 360 videos with continuous physiological measurements, subjective emotional ratings and motion traces," in *Proceedings of the 13th ACM Multimedia Systems Conference*, 2022, pp. 252–258.

[19] Y. Jin, J. Liu, F. Wang, and S. Cui, "Where are you looking? a large-scale dataset of head and gaze behavior for 360-degree videos and a pilot study," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1025–1034.