Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Viewport-Aware Immersive Video Content Identification Using Transformation-Robust Keypoints

Byeongchan Park

pbc866@ssu.ac.kr

Soongsil University

Research Article

Keywords: Immersive video, Content recognition, Robust feature points, Viewport-aware processing, Video fingerprinting

Posted Date: September 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11023835/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Viewport-Aware Immersive Video Content Identification Using Transformation-Robust Keypoints

Byeongchan Park[1]*

[1]Department of Computer Science and Engineering, Soongsil University, Seoul, Korea
Corresponding Author

Abstract
Immersive video content is frequently redistributed after compression, geometric modification, projection conversion, and viewport cropping. However, spatial distortion introduced by equirectangular projection and changes in the visible region reduce the repeatability of conventional local features, making transformed-content identification difficult. This study proposes a deep-learning-based immersive video recognition method using viewport-aware preprocessing and transformation-robust feature points. The proposed method selects representative frames, converts each immersive frame into multiple viewport regions, removes regions with insufficient visual information, and extracts object-centered feature points. Feature points that maintain positional and descriptive consistency between original and transformed content are retained as robust descriptors for reference indexing and query matching. The method was evaluated using 100 original immersive videos and 18,000 queries generated under 18 signal, geometric, temporal, and immersive-video-specific transformation conditions. The experimental results yielded a recognition rate of 97.5%, a misrecognition rate of 2.0%, and a non-recognition rate of 0.5%, with an average query response time of approximately 1.03 s. In the component-removal evaluation, full-frame matching without preprocessing achieved a recognition rate of 59.3% and an average response time of 5.58 s, whereas preprocessing without object-region extraction achieved 92.0% and 3.22 s, respectively. These results indicate that viewport-aware region processing and transformation-consistent feature selection improve both recognition robustness and computational efficiency. The proposed method can support immersive content retrieval, duplicate detection, and unauthorized redistribution monitoring without relying on filenames or embedded metadata.

Keywords Immersive video · Content recognition · Robust feature points · Viewport-aware processing · Video fingerprinting

1 Introduction

Immersive video provides a panoramic visual experience by capturing an entire spherical scene and allowing viewers to interactively select their viewing direction.

Its use has expanded across virtual reality, immersive broadcasting, cultural heritage, education, entertainment, and location-based media. As immersive content is increasingly distributed through online platforms, reliable content identification has become important for copyright management, duplicate detection, distribution tracking, and media asset retrieval.

Immersive video is commonly stored using an equirectangular projection (ERP), which maps a spherical scene onto a rectangular image plane. Although ERP facilitates conventional video encoding and transmission, its spatial distortion varies according to latitude. Visual regions near the poles are stretched more severely than equatorial regions, and visually identical scenes may produce different local appearances after spherical rotation or viewport conversion. Viewport selection, field-of-view modification, cropping, resolution conversion, compression, and temporal editing further change the observable content (Corbillon et al. 2017; Coors et al. 2018). Consequently, a content identification method designed for perspective video cannot be directly applied to immersive video without considering spherical geometry and viewport-dependent distortion.

Local feature methods such as SIFT, SURF, and ORB have been widely used for visual matching because they provide partial invariance to scale, rotation, and illumination changes (Lowe 2004; Bay et al. 2008; Rublee et al. 2011). However, these methods determine keypoints primarily from local image responses. They do not explicitly estimate whether the detected points will remain observable after ERP distortion, viewport conversion, cropping, recompression, or other combined transformations. A large number of unstable points may therefore be stored in the feature index, increasing both matching cost and false correspondence probability.

Learning-based methods, including SuperPoint, R2D2, DISK, ALIKED, and XFeat, have improved local feature detection and description by learning repeatable and discriminative representations from data (DeTone et al. 2018; Revaud et al. 2019; Tyszkiewicz et al. 2020; Zhao et al. 2023; Potje et al. 2024). Spherical convolution and projection-aware feature extraction have also been studied to reduce distortion in omnidirectional imagery (Cohen et al. 2018; Coors et al. 2018). Nevertheless, most existing methods focus on image correspondence, camera localization, or geometric reconstruction. They do not directly address the identification of transformed immersive video from a large reference collection.

Video copy detection methods generally identify content using global video representations, local descriptors, temporal sequences, or learned video similarity (Law-To et al. 2007; Kordopatis-Zilos et al. 2019). These methods can detect conventional transformations such as compression, resizing, frame deletion, and partial copying. However, immersive video introduces additional variations caused by spherical rotation, ERP distortion, viewport cropping, and field-of-view changes. Global representations may lose spatially localized evidence, whereas conventional local representations may generate inconsistent keypoints across different viewports.

Another limitation is that representative frame selection, viewport processing, keypoint extraction, and content matching are frequently optimized independently. Selecting frames solely from pixel differences may overreact to signal-level distortion, while extracting features from every frame produces considerable

temporal redundancy. Similarly, selecting viewport regions only according to object detection may exclude structurally distinctive regions that do not contain recognizable objects. Reliable immersive-video identification therefore requires these processes to be connected through a consistent representation.

This study proposes a viewport-aware immersive video content identification method using transformation-robust keypoints. Representative frames are first selected by combining RGB variation, semantic variation, and object information. Each selected ERP frame is converted into 12 perspective viewports using four horizontal and three vertical orientations. Structural variation and object-aware information are then used to retain informative viewport regions.

Candidate keypoints and descriptors are extracted from the selected regions using a shared feature encoder. During training, original and transformed viewport pairs are used to derive keypoint robustness targets from detection repeatability, positional consistency, and descriptor stability. The trained model predicts the expected transformation robustness of each keypoint from a single viewport. Only points satisfying the detector-confidence and robustness criteria are retained for content indexing and query matching.

The indexed descriptors are used to retrieve candidate reference videos. Descriptor correspondences are weighted according to the predicted robustness of both reference and query keypoints. The retrieved candidates are subsequently verified using spherical geometric consistency and temporal consistency. The final content identity is determined by combining descriptor, geometric, and temporal scores. A query is rejected as unknown content when the maximum similarity does not satisfy the identification threshold.

The principal contributions of this study are as follows.

1. A representative frame selection method that combines low-level temporal variation, semantic variation, and object information to reduce redundant feature extraction.
2. A viewport-aware representation that converts ERP frames into 4×3 perspective viewports and selects regions using structural and object-aware information.
3. A transformation-robust keypoint learning method that predicts keypoint reliability from repeatability, positional consistency, and descriptor stability without requiring manually annotated keypoint correspondences.
4. A content identification procedure that integrates robustness-weighted descriptor retrieval, spherical geometric verification, and temporal verification for complete and partial immersive-video queries.

The proposed method is evaluated under 18 transformation conditions, including signal-level, spatial, temporal, and immersive-video-specific transformations. Conventional and learning-based local feature methods—including SIFT, SURF, ORB, SuperPoint, R2D2, DISK, ALIKED, and XFeat—are used as baselines. Identification accuracy is assessed using Recall@1, Recall@5, mean average precision, and F1-score. Keypoint repeatability, matching accuracy, index size, and query processing time are also measured. Component contributions are analyzed

through ablation experiments involving representative frame selection, viewport processing, region selection, and robustness prediction.

The remainder of this paper is organized as follows. Section 2 reviews immersive-video representation, local feature extraction, and content identification methods. Section 3 presents the proposed method. Section 4 describes the datasets, transformations, baselines, and evaluation protocol. Section 5 reports and discusses the experimental results, and Section 6 concludes the paper.

2 Related Work

2.1 Video Representation and Transformations

Immersive video captures visual information in all viewing directions and represents the spherical scene in a two-dimensional format for compression, transmission, and storage. Equirectangular projection is widely used because it can be processed using conventional video codecs and is supported by the Omnidirectional Media Format [1], [2]. However, mapping a spherical scene onto a rectangular plane introduces nonuniform spatial distortion, particularly near the polar regions. Consequently, visual patterns corresponding to the same object may exhibit different shapes, scales, and spatial distributions depending on their spherical positions.

A user observes only a limited viewport of an immersive video at a given time. Therefore, viewport extraction and viewpoint movement change the visible region without modifying the identity of the underlying content. Previous studies investigated viewport-adaptive delivery to reduce the transmission of unobserved regions [1] and proposed spherical convolution to compensate for projection-dependent distortion [3]. SphereNet further modified convolutional sampling locations according to spherical geometry, improving object detection and classification in omnidirectional images [5]. Related standardization studies also defined viewport-based 3DoF and 3DoF+ representations for immersive media services [4], [6].

Immersive video may undergo conventional transformations such as compression, resolution change, frame-rate reduction, rotation, cropping, and photometric adjustment. It may also be affected by projection conversion, monoscopic or stereoscopic cropping, and base-viewpoint shifts that arise from immersive media processing. Previous research defined robustness indicators for evaluating immersive 360-degree video fingerprinting under these transformations [7]. However, the same feature detector was generally applied to entire projected frames, without explicitly separating regions according to viewport geometry or feature stability.

Most existing immersive media studies have focused on streaming efficiency, projection quality, visual saliency, or object recognition. Although spherical convolution methods reduce projection distortion, they do not directly determine whether an altered query video corresponds to registered reference content. Conversely, existing fingerprinting studies evaluate predefined transformations but provide limited mechanisms for selecting feature points that remain consistent

across both conventional and immersive-video-specific modifications. This limitation motivates the integration of representative-frame selection, viewport-aware region processing, and transformation-consistent feature-point selection in the proposed method.

2.2 Feature-Based Video Content Recognition

Feature-based video recognition identifies copied or transformed content by comparing visual information extracted from a query with descriptors registered in a reference database. Unlike filename- or metadata-based identification, visual feature matching can detect content after re-encoding, format conversion, or partial modification. Large-scale video-copy datasets such as VCDB have demonstrated the importance of local visual features for identifying partial and transformed video copies [8].

Because extracting features from every video frame requires substantial storage and processing time, representative-frame selection is commonly performed before feature extraction. Scene-change detection based on color differences or local features can reduce temporal redundancy while preserving visually distinguishable scenes [9]. Korean video-filtering standards have also defined recognition, misrecognition, and non-recognition measures for evaluating feature-based content identification [14]. For immersive video, RGB variation and object information have been jointly used to select representative frames [17]. However, representative-frame selection alone does not ensure that the extracted features remain stable after projection changes or viewport cropping.

SIFT detects extrema in scale space and generates descriptors that are robust to scale and rotation changes [10]. SURF reduces computational complexity by using integral images and approximated Hessian filters [12], while ORB combines oriented FAST keypoints with rotated BRIEF descriptors to provide efficient binary matching [13]. Comparative studies have shown that the selection between SIFT and SURF involves a trade-off between descriptor robustness and processing efficiency [11].

These handcrafted methods were originally designed for planar images. When directly applied to equirectangular frames, their local sampling regions are distorted according to latitude, causing inconsistent keypoint locations and descriptors. Spherical SIFT constructs scale-space representations in spherical coordinates to improve invariance for omnidirectional images [15]. SPHORB similarly extends binary feature extraction to the sphere and provides faster spherical feature matching [16].

Although spherical features compensate for geometric projection distortion, they generally extract features from broad image regions without considering content-specific importance. As a result, background textures, projection boundaries, or visually unstable regions may generate unnecessary candidate features. Furthermore, the detected keypoints are determined from an individual frame rather than their repeatability across expected content transformations. Therefore, conventional and spherical feature methods do not fully address the joint requirements of

representative-frame selection, viewport-aware region processing, transformation consistency, and efficient content-database matching.

2.3 Learning-Based Robust Feature Extraction

The limited adaptability of handcrafted detectors has motivated learning-based methods that jointly optimize keypoint detection and descriptor generation. Domestic feature-filtering studies have primarily evaluated robustness through recognition performance under predefined transformations [19], whereas recent learning-based approaches attempt to learn repeatable and discriminative local representations directly from training data.

SuperPoint jointly detects interest points and generates descriptors through a fully convolutional network. Its homographic adaptation procedure improves keypoint repeatability without requiring manually labeled interest points [18]. R2D2 extends this approach by separately estimating repeatability and reliability, thereby suppressing repeatable but ambiguous regions [20]. These methods demonstrate that the stability of a detected point and the discriminative capability of its descriptor should be considered together.

DISK learns local features through policy-gradient optimization and directly rewards correct feature correspondences [21]. ALIKED introduces a sparse deformable descriptor head to extract robust descriptors with reduced computational complexity [22]. More recently, XFeat was designed for lightweight sparse and semi-dense matching and provides an improved balance between matching accuracy and processing speed [23]. LightGlue further improves efficiency by adaptively reducing the number of keypoints and terminating the matching process according to image-pair difficulty [25].

These methods have mainly been evaluated using planar-image homography, camera-pose estimation, and visual-localization benchmarks. Their learned features are therefore not explicitly optimized for the nonuniform distortion of equirectangular projection, viewport-dependent visibility, or immersive-video-specific transformations. Direct application to an entire immersive frame may also retain features from distorted, low-information, or semantically irrelevant regions.

The proposed method addresses this gap without introducing a separate local-feature architecture. It combines viewport-aware projection processing with deep-learning-based object-region extraction and selects feature points according to their consistency across expected transformations. Consequently, the resulting descriptors are optimized for immersive video identification rather than general image correspondence. This combination distinguishes the proposed method from existing learning-based feature extractors and provides the basis for the recognition procedure described in Chapter 3.

3 Viewport-Aware Transformation-Robust Content Identification Method

3.1 Problem Formulation and Method Overview

The viewport-aware immersive video content identification method proposed in this study identifies the original content corresponding to a transformed query by combining representative frame selection, viewport-aware region processing, transformation-robust keypoint extraction, and verified feature matching. The overall configuration and processing procedure are shown in Fig. 1.

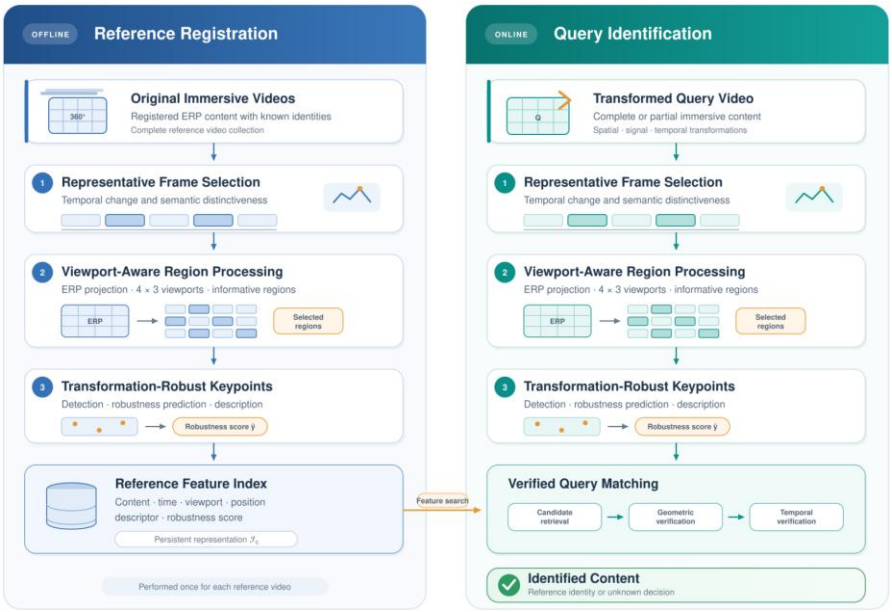


Fig. 1 Overall configuration and processing procedure of the proposed viewport-aware immersive video content identification method

As shown in Fig. 1, the proposed method consists of offline reference registration and online query identification. At the top, original immersive videos are provided for registration, whereas complete or partial transformed videos are received as queries.

In both procedures, representative frames are selected to reduce temporal redundancy. Each selected ERP frame is converted into 12 perspective viewports using four horizontal and three vertical orientations. Structurally distinctive and object-rich regions are retained, and candidate keypoints are assigned transformation-robustness scores. During offline registration, the retained descriptors and their metadata are stored in the reference feature index. During online identification, query descriptors retrieve candidate reference videos, which are subsequently verified using spherical geometric and temporal consistency.

Let the reference database contain N immersive videos. The reference database is defined as shown in Eq. (1).

$$\mathcal{D} = \{(c_n, V_n)\}_{n=1}^{N},\ V_n = \{I_{n,t}\}_{t=1}^{T_n}, \qquad (1)$$

where c_n denotes the identity of the n-th reference content, V_n denotes its frame sequence, $I_{n,t}$ is the frame at time t, and T_n is the total number of frames.

A query generated by applying one or more transformations to reference content is represented as shown in Eq. (2).

$$Q = \mathcal{T}(V_{c^*}; \xi) = \{\tilde{I}_\tau\}_{\tau=1}^{T_Q}, \qquad (2)$$

where c^* is the ground-truth query identity, $\mathcal{T}(\cdot)$ denotes the content transformation operator, and ξ represents its type and intensity. The operator includes signal-level, spatial, temporal, and immersive-video-specific transformations.

After representative frame selection, viewport conversion, and robust keypoint extraction, each indexed reference feature is represented as shown in Eq. (3).

$$e_{n,i} = (c_n, t_i, m_i, p_i, z_i, \hat{y}_i),\ \mathcal{I}_{c_n} = \{e_{n,i}\}_{i=1}^{M_n}, \qquad (3)$$

where t_i and m_i denote the temporal position and viewport index, respectively. In addition, p_i denotes the keypoint position, z_i denotes its descriptor, and $\hat{y}_i$ denotes its predicted transformation-robustness score. The reference feature index associated with c_n is denoted by $\mathcal{I}_{c_n}$.

The query feature representation generated using the same processing procedure is defined as shown in Eq. (4).

$$\Phi(Q) = \{e_j^Q\}_{j=1}^{M_Q},\ e_j^Q = (\tau_j, m_j^Q, p_j^Q, z_j^Q, \hat{y}_j^Q), \qquad (4)$$

where M_Q is the number of robust query keypoints and τ_j is the temporal position of query keypoint j.

The similarity $S(Q, c_n)$ combines robustness-weighted descriptor correspondence, spherical geometric consistency, and temporal consistency. The final content identity is determined as shown in Eq. (5).

$$\hat{c} = \begin{cases} \arg\max\limits_{c_n \in \mathcal{C}} S(Q, c_n), & \max\limits_{c_n \in \mathcal{C}} S(Q, c_n) \geq \tau_{\mathrm{id}}, \\ \text{unknown}, & \max\limits_{c_n \in \mathcal{C}} S(Q, c_n) < \tau_{\mathrm{id}}, \end{cases} \qquad (5)$$

where $\mathcal{C}$ is the set of registered content identities and τ_{id} is the identification threshold. A query is rejected as unknown when no reference content satisfies the threshold.

The principal notation used in the proposed method is summarized in Table 1.

Table 1 Principal notation used in the proposed method

Symbol	Description
$\mathcal{D}$	Reference immersive video database
N	Number of registered reference videos
c_n	Identity of the n-th reference content
$\mathcal{C}$	Set of registered content identities
V_n	Frame sequence of the n-th reference video
$I_{n,t}$	Frame at time $t$in V_n
T_n	Total number of frames in V_n
Q	Transformed query video
$\tilde{I}_\tau$	Frame at time τin query Q
$\mathcal{T}(\cdot)$	Content transformation operator
$\boldsymbol{\xi}$	Transformation type, intensity, and parameters
m_i	Viewport index of the i-th keypoint
$\boldsymbol{p}_i$	Spatial location of the i-th keypoint
$\boldsymbol{z}_i$	Descriptor of the i-th keypoint
$\hat{y}_i$	Predicted transformation-robustness score
$\mathcal{J}_{c_n}$	Feature index associated with content c_n
$\Phi(Q)$	Robust keypoint representation of query Q
$S(Q, c_n)$	Verified similarity between $Q$and c_n
τ_{id}	Content identification threshold
$\hat{c}$	Predicted content identity

Representative frame selection is described in Section 3.2, viewport-aware representation and region selection in Section 3.3, transformation-robust keypoint learning in Section 3.4, and content indexing and query identification in Section 3.5.

3.2 Representative Frame Selection

The representative frame selection procedure reduces temporal redundancy while retaining frames containing meaningful visual and semantic changes. Its overall configuration and processing procedure are shown in Fig. 2.

As shown in Fig. 2, the input video is processed through candidate frame sampling, RGB variation analysis, semantic variation analysis, object-aware scoring, integrated importance calculation, and temporal redundancy removal. The same procedure is applied to reference and query videos.

For an input video $V = \{I_t\}_{t=1}^{T}$, candidate frames are sampled at interval Δ_s. The candidate frame sequence is defined as shown in Eq. (6).

$$\mathcal{F}_s(V) = \left\{I_{t_k}, |, t_k, = 1 + k\Delta_s \ k = 0 \ldots K_s\right\},\ K_s = \left\lfloor \frac{T-1}{\Delta_s} \right\rfloor, \tag{6}$$

where $T$is the total number of frames and $K_s + 1$is the number of sampled frames. Uniform sampling reduces the cost of repeatedly analyzing visually similar adjacent frames.

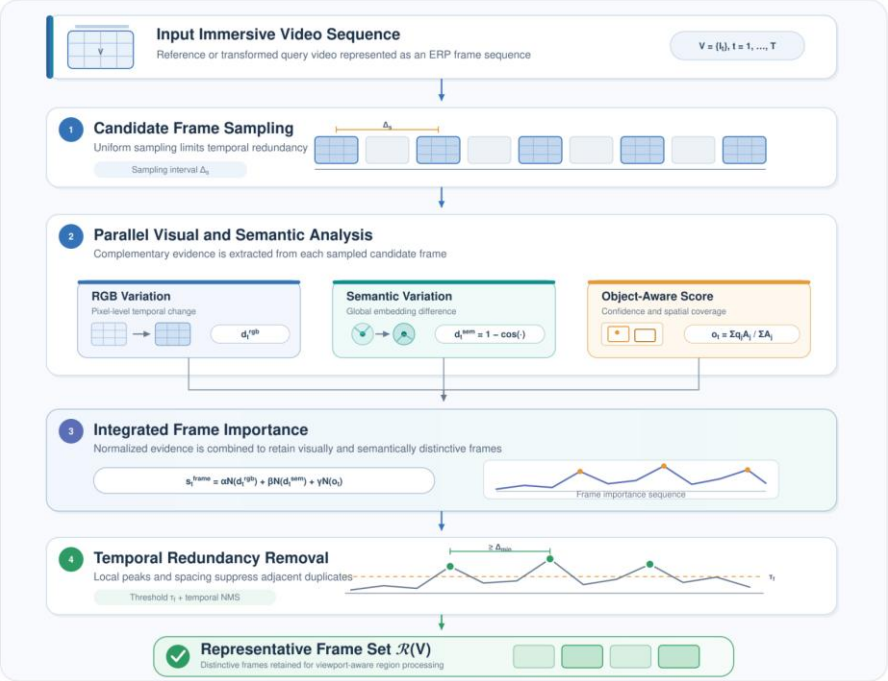


Fig. 2 Representative frame selection based on temporal variation and semantic information

The RGB variation between consecutive candidate frames is calculated as shown in Eq. (7).

$$d_{t_k}^{\mathrm{rgb}} = \frac{1}{3HW} \sum_{x=1}^{W} \sum_{y=1}^{H} \sum_{c\in\{R,G,B\}} \left| I_{t_k}(x,y,c) - I_{t_{k-1}}(x,y,c) \right|, \tag{7}$$

where W and H denote the frame width and height. A high value indicates a scene transition, camera movement, or substantial change in visual composition.

Because RGB differences can be affected by compression, brightness, and noise, semantic variation is additionally measured using global frame embeddings. The semantic variation score is calculated as shown in Eq. (8).

$$d_{t_k}^{\mathrm{sem}} = 1 - \frac{f_{t_k}^{T} f_{t_{k-1}}}{\left\| f_{t_k} \right\|_2 \left\| f_{t_{k-1}} \right\|_2 + \varepsilon}, \; f_{t_k} = g\left(I_{t_k}\right), \tag{8}$$

where $g(\cdot)$ denotes the semantic feature encoder and f_{t_k} is the global embedding of candidate frame I_{t_k}. This score responds to changes in scene context rather than only to pixel-level differences.

Object information is used to determine whether a frame contains semantically useful visual regions. The object-aware score is calculated as shown in Eq. (9).

$$o_{t_k} = \frac{\sum_{j=1}^{J_{t_k}} q_{t_k,j} A_{t_k,j}}{\sum_{j=1}^{J_{t_k}} A_{t_k,j} + \varepsilon}, \qquad (9)$$

where $J_{t_k}$is the number of detected objects, $q_{t_k,j}$is the confidence of object j, and $A_{t_k,j}$is its bounding-box area. The score is set to zero when no object is detected.

The three measurements are normalized and combined. The integrated frame importance is calculated as shown in Eq. (10).

$$s_{t_k}^{\text{frame}} = \alpha \mathcal{N}\left(d_{t_k}^{\text{rgb}}\right) + \beta \mathcal{N}\left(d_{t_k}^{\text{sem}}\right) + \gamma \mathcal{N}\left(o_{t_k}\right),\ \alpha + \beta + \gamma = 1, \qquad (10)$$

where $\mathcal{N}(\cdot)$denotes min–max normalization over the candidate frames. The weights α, β, and $\gamma$are determined using validation data and fixed for reference and query processing.

Frames satisfying the importance threshold and corresponding to local peaks are determined as shown in Eq. (11).

$$\mathcal{P} = \left\{ t_k \,\middle|\, s_{t_k}^{\text{frame}} \geq \tau_f \; s_{t_k}^{\text{frame}} = \max_{u \in \mathcal{N}_r(t_k)} s_u^{\text{frame}} \right\}, \qquad (11)$$

where τ_f is the frame importance threshold and $\mathcal{N}_r(t_k)$ is the temporal neighborhood around t_k. Local peak detection prevents multiple frames from being retained from the same gradual transition.

A minimum temporal distance is subsequently applied to remove adjacent redundant peaks. The final representative frame set is determined as shown in Eq. (12).

$$\mathcal{R}(V) = \{I_t \mid t \in \mathcal{T}_R\},\ \mathcal{T}_R = \sum_{t \in \mathcal{A}} s_t^{\text{frame}} \text{ subject to } \mid t_i - t_j \mid \geq \Delta_{\min}, \qquad (12)$$

where $\Delta_{\min}$ is the minimum temporal distance between two selected frames. Candidate peaks are examined in descending order of importance, and a frame is retained only when it satisfies the distance constraint. If no frame exceeds τ_f, the frame with the highest importance score is selected.

Only the frames contained in $\mathcal{R}(V)$ are transferred to the viewport-aware processing described in Section 3.3. This reduces redundant feature extraction while preserving visually and semantically distinctive content.

3.3 Viewport-Aware Representation and Region Selection

The viewport-aware processing procedure converts representative ERP frames into perspective viewports and retains regions expected to produce stable local features. Its overall configuration and processing procedure are shown in Fig. 3.

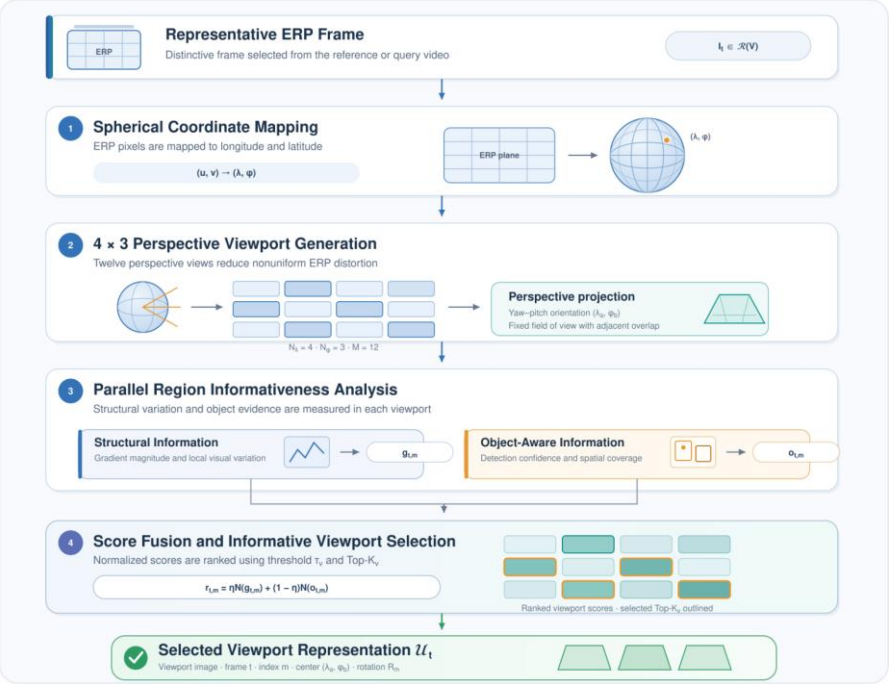


Fig. 3 Viewport-aware representation and informative region selection from an ERP frame

As shown in Fig. 3, each representative ERP frame is processed through spherical coordinate mapping, 4×3 perspective viewport generation, structural and object-aware analysis, and informative viewport selection. This reduces ERP distortion and excludes visually homogeneous regions before keypoint extraction.

For an ERP frame I_t with width W and height H, pixel position (u, v) is converted into spherical longitude and latitude as shown in Eq. (13).

$$\lambda(u) = 2\pi\left(\frac{u+0.5}{W}\right) - \pi,\ \phi(v) = \frac{\pi}{2} - \pi\left(\frac{v+0.5}{H}\right), \qquad (13)$$

where $\lambda \in [-, \pi\pi)$ and $\phi \in [-, \pi/2\pi/2]$. The horizontal position represents longitude, whereas the vertical position represents latitude.

The spherical scene is divided into four horizontal and three vertical orientations. The center orientation of viewport $m = (a, b)$ is determined as shown in Eq. (14).

$$\lambda_a = -\pi + \left(a + \frac{1}{2}\right)\frac{2\pi}{N_\lambda},\ \phi_b = \frac{\pi}{2} - \left(b + \frac{1}{2}\right)\frac{\pi}{N_\phi}, \qquad (14)$$

where $N_\lambda = 4$, $N_\phi = 3$, $a = 0, \ldots, 3,$ and $b = 0, \ldots, 2$. Consequently, $M = N_\lambda N_\phi = 12$ viewports are generated from each representative frame. Adjacent viewports partially overlap to prevent the loss of features located near viewport boundaries.

Let W_p, H_p, ω_h, and $\omega_v$denote the viewport width, height, horizontal field of view, and vertical field of view, respectively. Viewport pixel $(x, y)$is converted into a normalized camera ray as shown in Eq. (15).

$$\begin{aligned} x_p &= \left(\frac{2x}{W_p - 1} - 1\right)\tan\left(\frac{\omega_h}{2}\right), \\ y_p &= \left(1 - \frac{2y}{H_p - 1}\right)\tan\left(\frac{\omega_v}{2}\right), \\ r_p &= \frac{[x_p, y_p, 1]^T}{\left\|[x_p, y_p, 1]^T\right\|_2}. \end{aligned} \tag{15}$$

The camera ray is rotated according to the center orientation of viewport m. The corresponding spherical direction is calculated as shown in Eq. (16).

$$\begin{aligned} r_s &= R(\lambda_a, \phi_b) r_p = [X_s, Y_s, Z_s]^T, \\ \lambda_s &= \text{atan2}(X_s, Z_s),\ \phi_s = \arcsin(Y_s), \end{aligned} \tag{16}$$

where $R(\lambda_a, \phi_b)$denotes the yaw–pitch rotation matrix associated with viewport m.

The spherical direction is mapped back to the ERP frame to generate the perspective viewport. This mapping is defined as shown in Eq. (17).

$$\begin{aligned} u_s &= W\left(\frac{\lambda_s + \pi}{2\pi}\right) - \frac{1}{2}, \\ v_s &= H\left(\frac{\frac{\pi}{2} - \phi_s}{\pi}\right) - \frac{1}{2}, \\ P_{t,m}(x, y) &= I_t(u_s, v_s), \end{aligned} \tag{17}$$

where $P_{t,m}$is the m-th viewport generated from frame I_t. Bilinear interpolation is used for noninteger coordinates, and horizontal coordinates are circularly wrapped at the ERP boundary.

The informativeness of each viewport is evaluated using structural variation and object information. The structural information score is calculated as shown in Eq. (18).

$$g_{t,m} = \frac{1}{W_p H_p} \sum_{x=1}^{W_p} \sum_{y=1}^{H_p} \sqrt{G_{x,t,m}(x, y)^2 + G_{y,t,m}(x, y)^2}\,, \tag{18}$$

where $G_{x,t,m}$and $G_{y,t,m}$denote the horizontal and vertical gradients. A high value indicates that the viewport contains edges, corners, or local structures suitable for keypoint extraction.

The object-aware score of each viewport is calculated as shown in Eq. (19).

$$o_{t,m} = \min\left(1,\ \frac{\sum_{j=1}^{J_{t,m}} q_{t,m,j}\, A_{t,m,j}}{W_p H_p + \varepsilon}\right), \qquad (19)$$

where $J_{t,m}$is the number of detected objects, $q_{t,m,j}$is the confidence of object j, and $A_{t,m,j}$is its bounding-box area. The score is set to zero when no object is detected.

The structural and object-aware scores are normalized across the 12 viewports and combined. The integrated viewport score is calculated as shown in Eq. (20).

$$r_{t,m} = \eta \mathcal{N}\left(g_{t,m}\right) + (1-\eta)\mathcal{N}\left(o_{t,m}\right),\ 0 \le \eta \le 1, \qquad (20)$$

where ηcontrols the contribution of structural information. This combination retains texture-rich regions even when no object is detected while emphasizing semantically meaningful regions when object information is available.

Viewports satisfying the informativeness threshold are ranked, and a maximum of $K_v$viewports is retained. The selected viewport index set is determined as shown in Eq. (21).

$$\mathcal{M}_t^* = \begin{cases} \mathrm{TopK}_{K_v}\left(\{m \mid r_{t,m} \ge \tau_v\}; r_{t,m}\right), & \exists m: r_{t,m} \ge \tau_v, \\ \left\{\arg\max_m r_{t,m}\right\}, & \text{otherwise}, \end{cases} \qquad (21)$$

where $\tau_v$is the viewport selection threshold. If no viewport satisfies the threshold, the highest-scoring viewport is retained.

The selected viewport representation transferred to the keypoint extraction stage is defined as shown in Eq. (22).

$$\mathcal{U}_t = \left\{\left(P_{t,m}, t, m, \lambda_a, \phi_b, R_m\right) \middle|\ m \in \mathcal{M}_t^*\right\}. \qquad (22)$$

The frame position, viewport index, center orientation, and rotation matrix are retained to convert local keypoint positions into global spherical coordinates. Only the viewports contained in $\mathcal{U}_t$are transferred to the transformation-robust keypoint learning procedure described in Section 3.4.

3.4 Transformation-Robust Keypoint Learning

The proposed keypoint learning method predicts whether a local feature can be repeatedly detected and correctly matched after content transformation. It evaluates candidate points using detection repeatability, positional consistency, and descriptor stability. The overall training and inference procedures are shown in Fig. 4.

As shown in Fig. 4, the training procedure consists of transformed viewport generation, shared feature encoding, transformation-guided correspondence construction, robustness target generation, and multi-task learning. During inference, a single viewport is processed, and only keypoints satisfying the detector-confidence and robustness criteria are retained.

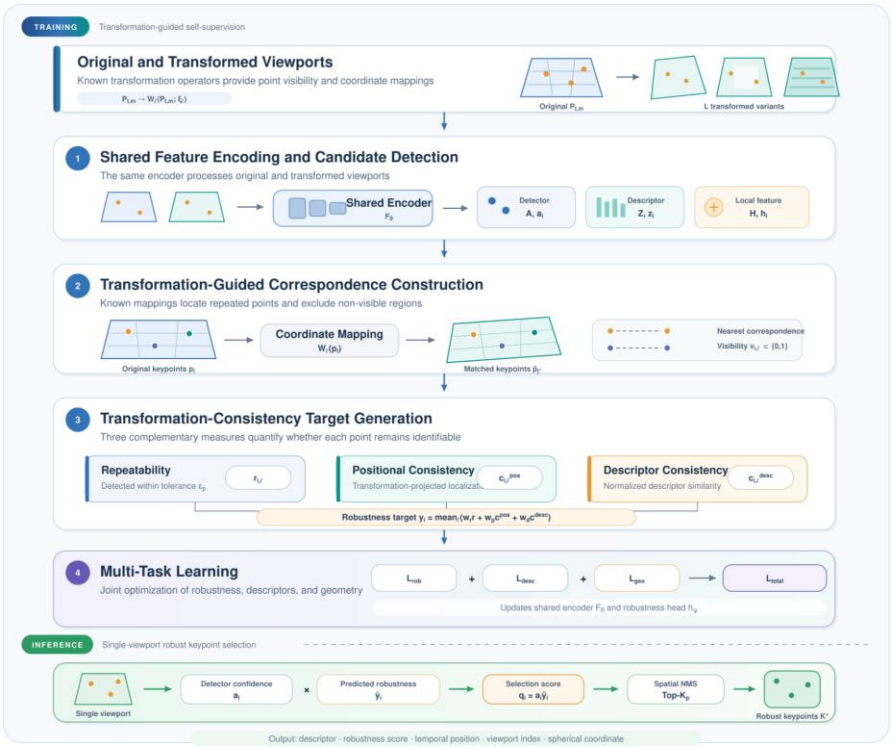


Fig. 4 Transformation-robust keypoint learning using repeatability, positional consistency, and descriptor stability

For a selected viewport $P_{t,m} \in \mathcal{U}_t$, transformed training views and the visibility of candidate point $p_i$are defined as shown in Eq. (23).

$$\begin{aligned} \tilde{P}_{t,m}^{(\ell)} &= \mathcal{W}_\ell\left(P_{t,m}; \xi_\ell\right),\ \ell = 1, \dots, L, \\ v_{i,\ell} &= \mathbb{I}\left[\mathcal{W}_\ell(p_i) \in \Omega\left(\tilde{P}_{t,m}^{(\ell)}\right)\right], \end{aligned} \tag{23}$$

where $\mathcal{W}_\ell(\cdot)$denotes a transformation operator, $\xi_\ell$denotes its parameters, and $L$is the number of transformed views. The visibility indicator $v_{i,\ell}$ excludes points removed by cropping, occlusion, or spatial loss from the consistency calculation.

The transformations include compression, brightness variation, blur, noise, scaling, rotation, cropping, perspective deformation, and field-of-view modification. Their known coordinate mappings enable correspondence labels to be generated without manually annotated keypoints.

The original and transformed viewports are processed using the same feature encoder. Candidate keypoint extraction and robustness prediction are represented as shown in Eq. (24).

$$\begin{aligned} \mathcal{K}\left(P_{t,m}\right) &= F_\theta\left(P_{t,m}\right) = \{(p_i, a_i, z_i, h_i)\}_{i=1}^{M_{t,m}}, \\ \hat{y}_i &= \sigma\left(h_\psi(h_i)\right), \end{aligned} \tag{24}$$

where $a_i$denotes detector confidence, $z_i$denotes the local descriptor, and $h_i$is the local representation used by the robustness head $h_\psi(\cdot)$. The predicted robustness score is denoted by $\hat{y}_i$.

For each visible original keypoint, the nearest keypoint in the transformed viewport is located using the known coordinate mapping. The correspondence and repeatability are determined as shown in Eq. (25).

$$\begin{aligned} j^*(i,\ell) &= \left\| \mathcal{W}_\ell(p_i) - \tilde{p}_j^{(\ell)} \right\|_2, \\ e_{i,\ell}^{\text{pos}} &= \left\| \mathcal{W}_\ell(p_i) - \tilde{p}_{j^*(i,\ell)}^{(\ell)} \right\|_2, \\ r_{i,\ell} &= v_{i,\ell} \mathbb{I}\left[e_{i,\ell}^{\text{pos}} \le \varepsilon_p \right], \end{aligned} \tag{25}$$

where $\varepsilon_p$is the allowable localization error. A point is repeatable when it remains visible and is detected within the positional tolerance.

Positional and descriptor consistency are calculated as shown in Eq. (26).

$$\begin{aligned} c_{i,\ell}^{\text{pos}} &= r_{i,\ell} \exp\left(-\frac{\left(e_{i,\ell}^{\text{pos}}\right)^2}{2\sigma_p^2} \right), \\ c_{i,\ell}^{\text{desc}} &= r_{i,\ell} \frac{1 + \text{sim}\left(z_i, \tilde{z}_{j^*(i,\ell)}^{(\ell)} \right)}{2}, \end{aligned} \tag{26}$$

where $\sigma_p$controls sensitivity to positional error and $\text{sim}(\cdot)$denotes cosine descriptor similarity. Both scores are set to zero when the point is not repeatedly detected.

The transformation-robustness target is obtained by combining the three consistency measures across the visible transformations. The target is calculated as shown in Eq. (27).

$$y_i = \frac{\sum_{\ell=1}^{L} v_{i,\ell} \left(w_r r_{i,\ell} + w_p c_{i,\ell}^{\text{pos}} + w_d c_{i,\ell}^{\text{desc}} \right)}{\sum_{\ell=1}^{L} v_{i,\ell} + \varepsilon}, \; w_r + w_p + w_d = 1, \tag{27}$$

where $y_i \in [0,1]$represents the expected stability of candidate keypoint i. A high target indicates that the point remains detectable, spatially consistent, and descriptively similar under multiple transformations.

The robustness head is trained using the soft-label binary cross-entropy loss defined as shown in Eq. (28).

$$\mathcal{L}_{\text{rob}} = -\frac{1}{M_{t,m}} \sum_{i=1}^{M_{t,m}} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right]. \tag{28}$$

Descriptor discrimination is optimized using an InfoNCE loss applied to geometrically valid keypoint pairs. A smooth L_1loss is additionally applied to the localization error between transformation-projected points and their detected correspondences. The complete training objective is calculated as shown in Eq. (29).

$$\mathcal{L}_{\text{total}} = \lambda_{\text{rob}} \mathcal{L}_{\text{rob}} + \lambda_{\text{desc}} \mathcal{L}_{\text{InfoNCE}} + \lambda_{\text{geo}} \mathcal{L}_{\text{SmoothL1}}, \tag{29}$$

where the three coefficients control robustness prediction, descriptor learning, and geometric consistency. Their values are determined using validation data.

At inference, transformed viewport pairs are not required. The trained model predicts keypoints, descriptors, and robustness scores from a single reference or query viewport. Detector confidence and predicted robustness are combined, and the final keypoint set is determined as shown in Eq. (30).

$$\begin{aligned} q_i &= a_i \hat{y}_i, \\ \mathcal{K}_{t,m}^{*} &= \mathrm{TopK}_{K_p}\left[\mathrm{NMS}_{r_{\mathrm{nms}}}(\{(p_i, z_i, \hat{y}_i) \mid q_i \geq \tau_k\})\right], \end{aligned} \tag{30}$$

where $q_i$denotes the keypoint selection score, $\tau_k$is the selection threshold, $r_{\mathrm{nms}}$is the spatial non-maximum suppression radius, and $K_p$is the maximum number of retained keypoints.

Finally, each selected keypoint is converted from a local viewport coordinate to a global spherical coordinate using the viewport rotation matrix. The conversion is performed as shown in Eq. (31).

$$\begin{aligned} r_{s,i} &= R_m \frac{\left[x_{p,i}, y_{p,i}, 1\right]^T}{\left\|\left[x_{p,i}, y_{p,i}, 1\right]^T\right\|_2} = [X_i, Y_i, Z_i]^T, \\ p_i^{\mathrm{sph}} &= \left(\mathrm{atan2}, (X_i, Z_i)\arcsin(Y_i)\right). \end{aligned} \tag{31}$$

The resulting robust keypoints retain their descriptors, robustness scores, temporal positions, viewport indices, and spherical coordinates. These features are transferred to the indexing and query identification procedure described in Section 3.5.

3.5 Similarity Search and Multi-Frame Identification

The proposed identification procedure retrieves candidate reference videos using transformation-robust descriptors and verifies them using spherical geometric and temporal consistency. Its overall configuration and processing procedure are shown in Fig. 5.

As shown in Fig. 5, the reference keypoints are stored in an offline feature index, whereas query keypoints are generated online using the procedures described in Sections 3.2–3.4. Both inputs are combined during descriptor retrieval, followed by spherical geometric verification, temporal verification, score integration, and identity determination.

Keypoints detected in overlapping viewports are first merged when their spherical distance is below a predefined threshold and their descriptors are sufficiently similar. The keypoint with the highest selection score is retained among duplicate points.

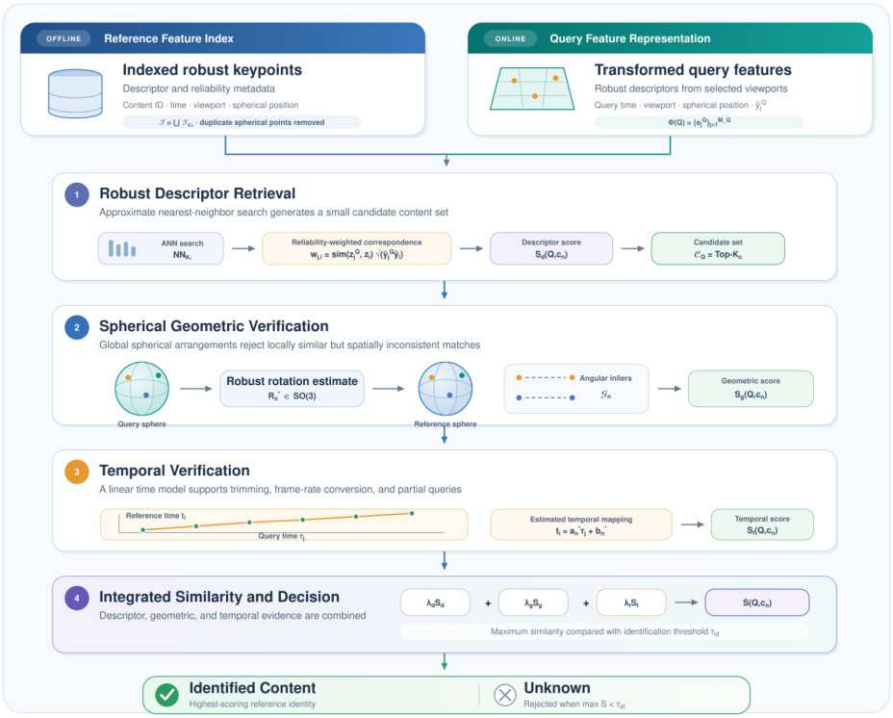


Fig. 5 Content indexing and query identification using descriptor retrieval, spherical geometric verification, and temporal verification

The indexed feature entry and complete reference feature index are defined as shown in Eq. (32).

$$\begin{aligned} e_{n,i} &= \left(c_n, t_i, m_i, p_{n,i}^{\mathrm{sph}}, \bar{z}_{n,i}, \hat{y}_{n,i}\right), \\ \bar{z}_{n,i} &= \frac{z_{n,i}}{\left\|z_{n,i}\right\|_2 + \varepsilon}, \\ \mathcal{I} &= \bigcup_{n=1}^{N} \mathcal{I}_{c_n}, \ \mathcal{I}_{c_n} = \left\{e_{n,i}\right\}_{i=1}^{M_n}, \end{aligned} \tag{32}$$

where c_n, t_i, and m_i denote the content identity, temporal position, and viewport index, respectively. In addition, $p_{n,i}^{\mathrm{sph}}$ is the spherical position, $\bar{z}_{n,i}$ is the normalized descriptor, and $\hat{y}_{n,i}$ is the predicted robustness score.

The query representation $\Phi(Q)$ contains the same information except for the unknown content identity. At the second level of Fig. 5, each query descriptor is searched against $\mathcal{I}$ using approximate nearest-neighbor retrieval. The descriptor correspondence set and its reliability weight are determined as shown in Eq. (33).

$$\mathcal{M}_n = \left\{ (j,i) \middle| \begin{array}{l} e_{n,i} \in \mathrm{NN}_{K_r}\left(\bar{z}_j^Q, \mathcal{I}\right), \\ \mathrm{sim}\left(\bar{z}_j^Q, \bar{z}_{n,i}\right) \geq \tau_d \end{array} \right\},$$
$$w_{j,i} = \max\left(0, \bar{z}_j^{QT}\, \bar{z}_{n,i}\right) \sqrt{\hat{y}_j^Q \hat{y}_{n,i}}, \tag{33}$$

where K_r is the number of retrieved neighbors and τ_d is the descriptor threshold. The weight $w_{j,i}$ decreases when either the reference or query keypoint has low predicted robustness.

The descriptor retrieval score and candidate content set are determined as shown in Eq. (34).

$$S_d(Q, c_n) = \frac{\sum_{(j,i)\in\mathcal{M}_n} w_{j,i}}{\sum_{j=1}^{M_Q} \hat{y}_j^Q + \varepsilon},$$
$$\mathcal{C}_Q = \mathrm{TopK}_{K_c}\left(\{c_n \mid S_d(Q, c_n) \geq \tau_c\}; S_d(Q, c_n)\right), \tag{34}$$

where K_c denotes the maximum number of candidate videos and τ_c denotes the candidate retrieval threshold. Geometric and temporal verification are performed only for content identities in $\mathcal{C}_Q$.

At the third level of Fig. 5, the spherical arrangement of the matched keypoints is verified. A spherical position (λ, ϕ) is represented by the unit vector

$$r(\lambda, \phi) = [\cos\phi\, , \sin\lambda\, ,\ \sin\phi\ \ \cos\phi\cos\lambda]^T.$$

For each candidate content, the rotation producing the largest weighted geometric consensus is estimated. The rotation, geometric inlier set, and geometric score are determined as shown in Eq. (35).

$$R_n^* = \sum_{(j,i)\in\mathcal{M}_n} w_{j,i}\, \mathbb{I}\left[d_{\mathbb{S}^2}\left(R, r_j^Q r_{n,i}\right) \leq \varepsilon_g\right],$$
$$\mathcal{G}_n = \left\{(j,i) \in \mathcal{M}_n \mid d_{\mathbb{S}^2}\left(R_n^*, r_j^Q r_{n,i}\right) \leq \varepsilon_g\right\},$$
$$S_g(Q, c_n) = \frac{\sum_{(j,i)\in\mathcal{G}_n} w_{j,i}}{\sum_{(j,i)\in\mathcal{M}_n} w_{j,i} + \varepsilon}, \tag{35}$$

where $d_{\mathbb{S}^2}(\cdot)$ is spherical angular distance and ε_g is the geometric tolerance. This stage removes descriptor matches that are inconsistent with the global spherical structure.

At the fourth level of Fig. 5, temporal verification determines whether the geometric inliers follow a common temporal scale and offset. The temporal model supports partial queries, trimming, and frame-rate conversion. Its parameters, inlier set, and verification score are determined as shown in Eq. (36).

$$
\begin{aligned}
(a_n^*, b_n^*) &= \sum_{(j,i)\in\mathcal{G}_n} w_{j,i}\, \mathbb{I}\big[|t_i - (a\tau_j + b)| \le \varepsilon_t\big], \\
\mathcal{H}_n &= \{(j,i) \in \mathcal{G}_n \mid |t_i - (a_n^*\tau_j + b_n^*)| \le \varepsilon_t\}, \\
S_t(Q, c_n) &= \frac{\sum_{(j,i)\in\mathcal{H}_n} w_{j,i}}{\sum_{(j,i)\in\mathcal{G}_n} w_{j,i} + \varepsilon},
\end{aligned}
\tag{36}
$$

where $a_n^*$and b_n^*denote the estimated temporal scale and offset, respectively, and $\varepsilon_t$is the temporal tolerance.

At the fifth level of Fig. 5, the three scores are combined. The final content similarity is calculated as shown in Eq. (37).

$$
S(Q, c_n) = \lambda_d S_d(Q, c_n) + \lambda_g S_g(Q, c_n) + \lambda_t S_t(Q, c_n),\ \lambda_d + \lambda_g + \lambda_t = 1, \tag{37}
$$

where λ_d, λ_g, and $\lambda_t$control the contributions of descriptor, geometric, and temporal evidence. Their values are determined using validation data.

The roles of the matching stages are summarized in Table 2.

Table 2 Matching and verification stages used for content identification

Stage	Primary evidence	Score	Acceptance or control parameter	Purpose
Descriptor retrieval	Robust descriptor similarity	S_d	τ_d, τ_c, K_r, K_c	Retrieve a small candidate content set
Geometric verification	Spherical keypoint arrangement	S_g	Angular tolerance ε_g	Remove spatially inconsistent descriptor matches
Temporal verification	Temporal scale, offset, and ordering	S_t	Temporal tolerance ε_t	Verify partial, trimmed, or resampled queries
Final identification	Combined retrieval and verification evidence	S	Identification threshold τ_{id}	Return a content identity or reject the query

The final identity is determined using Eq. (5). The highest-scoring content is returned when its similarity exceeds τ_{id}; otherwise, the query is rejected as unknown. Because geometric and temporal verification are limited to $K_c$candidates, the method maintains practical query processing time as the reference database increases.

4 Experimental Results

4.1 Datasets and Experimental Setup

The experimental environment consisted of a comparison search server and a comparison searcher. The overall configuration and data flow are shown in Fig. 6.

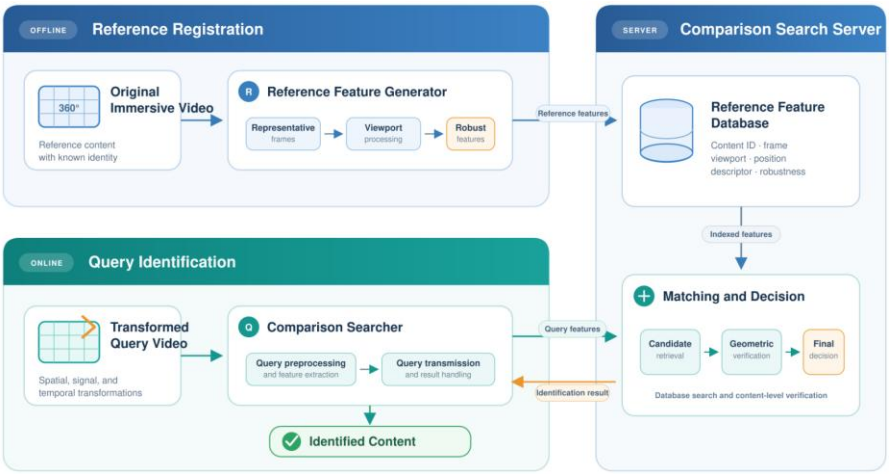


Fig. 6 Experimental environment for reference feature generation and immersive video content identification.

As shown in Fig. 6, the original immersive videos are first processed by the original-data generator. Representative frames are extracted and preprocessed, after which robust feature points are generated and stored in the reference database. For identification, the comparison searcher receives a transformed query video, extracts its query features, and sends them to the comparison search server. The server compares the query features with the indexed reference features and returns the corresponding content identifier.

The hardware specifications used for reference feature generation and query identification are summarized in Table 3.

Table 3 Hardware specifications of the experimental environment

Component	Comparison search server	Comparison searcher
CPU	Intel Core i9-12900K, 16 cores and 24 threads	AMD Ryzen 9 7950X, 16 cores and 32 threads
RAM	32 GB	32 GB
Storage	4 TB SATA × 3	1 TB SSD
GPU	NVIDIA GeForce RTX 4080	NVIDIA GeForce RTX 4090
Operating system	Windows 11 Pro 64-bit	Windows 11 Pro 64-bit

The server performed representative-frame extraction, preprocessing, robust feature-point generation, reference indexing, and database matching. The comparison searcher processed the query videos and transmitted the extracted query features to the server. This separation allowed the reference data to be generated in advance while each transformed query was evaluated independently.

The reference dataset consisted of 100 immersive videos collected from publicly accessible videos on YouTube and Vimeo. Representative examples of the original videos are shown in Fig. 7.

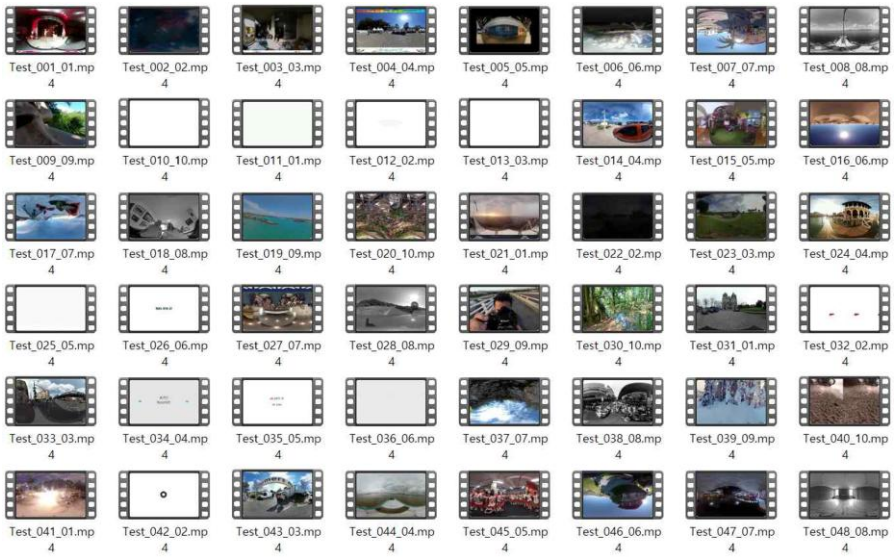


Fig. 7 Examples of the original immersive videos used to construct the reference dataset.

A shown in Fig. 7, the dataset includes immersive videos containing various scenes, objects, spatial structures, and camera movements. Each video was assigned a unique content identifier and processed to construct the reference feature database. The query dataset was generated by applying 18 transformation types to the original videos. For each transformation type, 1,000 query videos were generated, resulting in a total of 18,000 queries. The query set was used to evaluate content recognition, misrecognition, non-recognition, consistency, and processing speed. The individual transformation conditions and the number of queries are described in Sect. 4.2.

4.2 Transformation Conditions and Evaluation Metrics

The original immersive videos were modified using both conventional video transformations and immersive-video-specific transformations. Representative examples of the transformations are shown in Fig. 8.

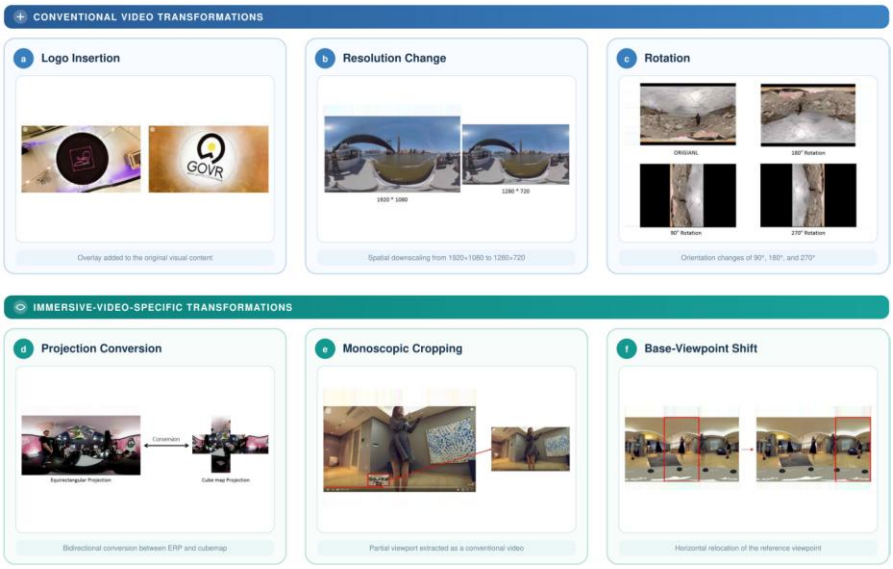


Fig. 8 Representative transformations applied to immersive video content: (a) logo insertion, (b) resolution change, (c) rotation, (d) projection conversion, (e) monoscopic cropping, and (f) base-viewpoint shift.

As shown in Fig. 8, the conventional transformations modify the visual quality, spatial structure, or playback characteristics of the video. In comparison, projection conversion, viewport cropping, and base-viewpoint shift alter the spherical representation or observable region of immersive content. These transformations were used to examine whether the proposed method could preserve content identity under different distortion conditions.

The complete transformation protocol is summarized in Table 4. Each transformation type contained 1,000 query videos, resulting in 18,000 queries.

Transformations 1–14 represent general modifications that can occur during content editing, transcoding, storage, and redistribution. The combined transformation applied at least two operations among compression, codec conversion, and caption insertion. File-level modification included archive compression and executable-file modification. Transformations 15–18 were designed specifically for immersive video and included ERP-to-CMP or CMP-to-ERP conversion, partial viewport extraction, and viewpoint movement.

The evaluation measures used to assess robustness, consistency, and efficiency are summarized in Table 5.

Table 4 Transformation conditions and number of query videos

No.	Transformation condition	Number of queries
1	Logo insertion	1,000
2	Caption insertion	1,000
3	Severe compression	1,000
4	Codec conversion	1,000
5	Aspect-ratio change	1,000
6	Resolution change	1,000
7	Frame-rate reduction	1,000
8	Rotation	1,000
9	Flip	1,000
10	Grayscale conversion	1,000
11	Brightness adjustment	1,000
12	Contrast adjustment	1,000
13	Combined transformation	1,000
14	File-level modification	1,000
15	Projection conversion	1,000
16	Monoscopic cropping	1,000
17	Stereoscopic cropping	1,000
18	Base-viewpoint shift	1,000
	Total	**18,000**

Table 5 Evaluation measures for immersive video content identification

Evaluation dimension	Measure	Evaluation procedure
Robustness	Recognition rate	Correct source-content response
	Misrecognition rate	Incorrect content response
	Non-recognition rate	No content response
Consistency	Feature extraction consistency	Repeated feature extraction
	Recognition consistency	Repeated identification
Efficiency	Average response time	Query input to identification response

As summarized in Table 5, robustness was evaluated using the distribution of correct recognition, misrecognition, and non-recognition results. Consistency was measured through repeated feature extraction and identification, while efficiency was measured using the average response time.

Let N denote the total number of queries, and let N_c, N_m, and N_u denote the numbers of correctly recognized, misrecognized, and unrecognized queries, respectively. The three response rates are calculated as shown in Eq. (38).

$$R_c = \frac{N_c}{N} \times 100,\ R_m = \frac{N_m}{N} \times 100,\ R_u = \frac{N_u}{N} \times 100 \qquad (38)$$

In Eq. (38), R_c, R_m, and $R_u$represent the recognition, misrecognition, and non-recognition rates, respectively, where $N = N_c + N_m + N_u$.

Feature extraction consistency was calculated using the proportion of repeatedly matched feature points, as shown in Eq. (39).

$$C_{\text{ext}} = \frac{N_{\text{matched}}^{\text{kp}}}{N_{\text{extracted}}^{\text{kp}}} \times 100 \qquad (39)$$

In Eq. (39), $N_{\text{matched}}^{\text{kp}}$is the number of feature points consistently matched across repeated extractions, and $N_{\text{extracted}}^{\text{kp}}$is the total number of extracted feature points.

Recognition consistency was calculated from repeated identification trials, as shown in Eq. (40).

$$C_{\text{rec}} = \frac{N_{\text{consistent}}}{N_{\text{repeat}}} \times 100 \qquad (40)$$

In Eq. (40), $N_{\text{consistent}}$denotes the number of trials producing the same content-identification result, and N_{repeat}denotes the total number of repeated trials.

Finally, the average response time was calculated as shown in Eq. (41).

$$\bar{t} = \frac{1}{N}\sum_{i=1}^{N}\left(t_i^{\text{end}} - t_i^{\text{start}}\right) \qquad (41)$$

In Eq. (41), $t_i^{\text{start}}$and t_i^{end}denote the query input time and the identification response time for the i-th query, respectively.

4.3 Registered and Unregistered Video Identification

The identification experiment was conducted using 18,000 transformed query videos, with 1,000 queries for each transformation condition. The recognition, misrecognition, and non-recognition results are summarized in Table 6.

As shown in Table 6, 17,550 of the 18,000 queries were correctly identified, corresponding to an overall recognition rate of 97.5%. Misrecognition occurred for 360 queries, whereas 90 queries produced no identification result. The corresponding misrecognition and non-recognition rates were 2.0% and 0.5%, respectively.

Table 6 Content identification results under the 18 transformation conditions

No.	Transformation condition	Recognition (%)	Misrecognition (%)	Non-recognition (%)
1	Logo insertion	100	0	0
2	Caption insertion	99	0	1
3	Severe compression	99	0	1
4	Codec conversion	99	0	1
5	Aspect-ratio change	100	0	0
6	Resolution change	99	1	0
7	Frame-rate reduction	98	2	0
8	Rotation	95	3	2
9	Flip	95	5	0
10	Grayscale conversion	97	3	0
11	Brightness adjustment	97	3	0
12	Contrast adjustment	97	3	0
13	Combined transformation	99	0	1
14	File-level modification	100	0	0
15	Projection conversion	100	0	0
16	Monoscopic cropping	92	6	2
17	Stereoscopic cropping	89	10	1
18	Base-viewpoint shift	100	0	0
Total	18 transformation conditions	97.5	2.0	0.5

Logo insertion, aspect-ratio change, file-level modification, projection conversion, and base-viewpoint shift achieved recognition rates of 100%. Logo insertion had limited influence because the upper and lower regions containing overlays were excluded during viewport preprocessing. Projection conversion also maintained full recognition because cubemap content was normalized to the ERP representation before feature extraction.

Caption insertion, compression, codec conversion, resolution change, and combined transformation achieved recognition rates of 99%. These results indicate that most local structures remained available for feature matching after signal-level degradation. Frame-rate reduction achieved 98%, while grayscale, brightness, and contrast transformations each achieved 97%. The local geometric structures used by the feature descriptors were generally preserved despite changes in color and luminance.

Rotation and flipping produced recognition rates of 95%. Their relatively lower results were associated with changes in feature orientation and spatial arrangement. Monoscopic and stereoscopic cropping showed the lowest recognition rates of 92% and 89%, respectively. Cropping removed substantial portions of the spherical content and reduced the number of objects and robust feature points shared with the reference video.

Representative feature-matching results are shown in Fig. 9.

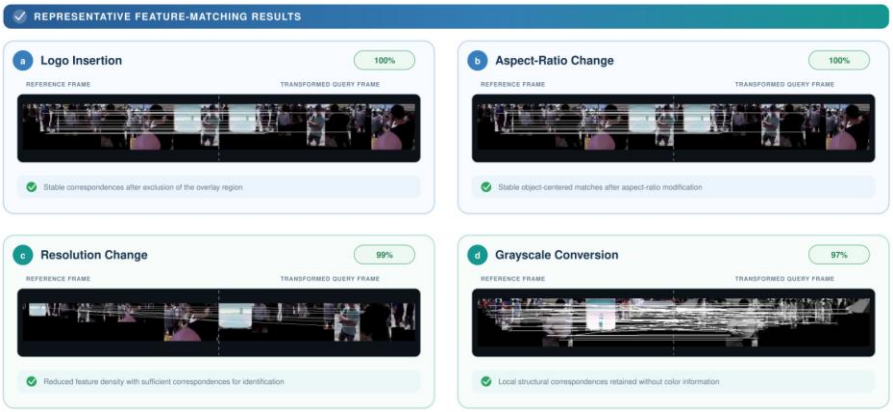


Table 7 Repeatability of content-identification results

As shown in Fig. 9, the proposed preprocessing retained stable object regions and local features under logo insertion and aspect-ratio modification. Resolution reduction decreased the number of extracted features but preserved sufficient correspondences for correct identification. Grayscale conversion changed the signal characteristics while retaining the local structures required for feature matching.

The repeatability of the identification results is summarized in Table 7. Four correctly recognized queries, three misrecognized queries, and three unrecognized queries were each evaluated 200 times.

Table 7 Repeatability of content-identification results

Initial result	Number of queries	Repetitions per query	Total trials	Consistent results (%)
Recognition	4	200	800	100
Misrecognition	3	200	600	100
Non-recognition	3	200	600	100
Total	**10**		**2,000**	**100**

As shown in Table 7, all 2,000 repeated trials produced the same outputs as their initial trials, resulting in 100% identification repeatability. This result indicates that the feature extraction and matching procedures generated deterministic responses under identical input conditions. However, repeatability represents output consistency and does not imply that repeatedly misrecognized or unrecognized queries were correctly identified.

4.4 Preprocessing Ablation and Efficiency Analysis

The computational efficiency of the proposed method was evaluated using 1,000 query videos for each transformation condition. The average response times measured across the 18 conditions are summarized in Table 8.

Table 8. Average response time under the 18 transformation conditions

No.	**Transformation**	**Time (ms)**	**No.**	**Transformation**	**Time (ms)**
1	Logo insertion	839	10	Grayscale conversion	714
2	Caption insertion	1,314	11	Brightness adjustment	714
3	Severe compression	673	12	Contrast adjustment	714
4	Codec conversion	2,813	13	Combined transformation	1,826
5	Aspect-ratio change	839	14	File-level modification	839
6	Resolution change	1,314	15	Projection conversion	839
7	Frame-rate reduction	679	16	Monoscopic cropping	835
8	Rotation	879	17	Stereoscopic cropping	973
9	Flip	872	18	Base-viewpoint shift	839
	Overall average	1,028.6			

As shown in Table 8, the proposed method required an average of 1,028.6 ms to identify a query video. Severe compression produced the shortest response time of 673 ms, whereas codec conversion required the longest time of 2,813 ms. The higher latency under codec conversion and combined transformation was caused by the increased number of candidate features retained during the matching process. Nevertheless, the average response time remained close to one second across all 18,000 queries.

To examine the contribution of the preprocessing components, three configurations were compared. Configuration A represents the complete proposed method. Configuration B extracts and matches features from entire frames without the preprocessing procedures. Configuration C retains the preprocessing procedures but excludes the final object-region extraction step. The definitions and overall results of these configurations are presented in Table 9.

Table 9. Overall comparison of the proposed method and preprocessing ablations

Configuration	Applied processing	Recognition rate (%)	Average response time (ms)
A	Complete proposed method	**97.5**	**1,029**
B	Full-frame matching without preprocessing	59.3	5,577
C	Preprocessing without object-region extraction	92.0	3,217

Table 9 shows that removing all preprocessing procedures reduced the recognition rate from 97.5% to 59.3% and increased the response time to 5,577 ms. Excluding only object-region extraction resulted in a smaller reduction in recognition performance, but the average response time still increased to 3,217 ms. These results indicate that viewport-aware preprocessing limits the search space, while object-region extraction removes visually irrelevant features before matching.

The overall effects of the preprocessing components on recognition performance and response time are visualized in Fig. 10.

As shown in Fig. 10, Configuration A achieved both the highest recognition rate and the lowest response time. Configuration B produced the largest performance degradation because features extracted from irrelevant or highly distorted regions increased false correspondences. Configuration C performed better than Configuration B, demonstrating the contribution of representative-frame selection and viewport-aware region processing even when object-region extraction was excluded.

The relative latency reduction achieved by Configuration A over an ablated configuration was calculated as defined in Eq. (42).

$$\Delta_t(A, X) = \frac{\bar{t}_X - \bar{t}_A}{\bar{t}_X} \times 100,\ X \in \{B, C\} \qquad (42)$$

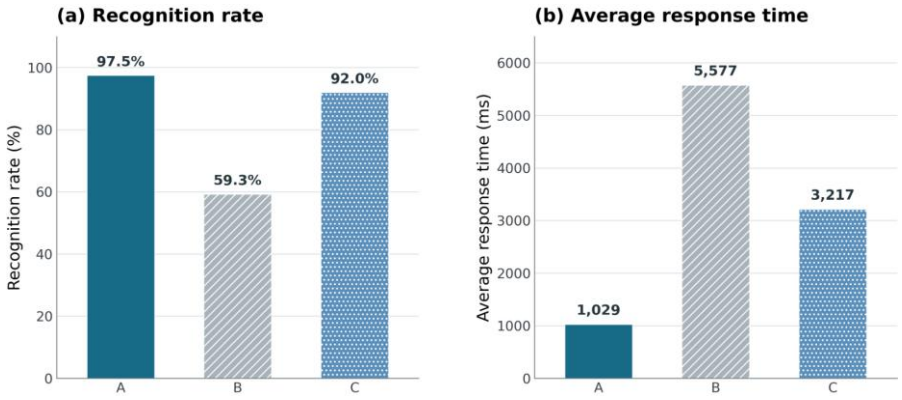


Fig. 10. Overall effects of preprocessing components on recognition rate and average response time.

where $\bar{t}_A$ denotes the average response time of the complete method and $\bar{t}_X$represents that of Configuration B or C. According to Eq. (42), Configuration A reduced the average latency by 81.6% relative to Configuration B and by 68.0% relative to Configuration C. This corresponds to approximately 5.4 and 3.1 times faster processing, respectively.

One exception was observed for monoscopic cropping, for which Configuration C achieved a recognition rate of 95%, compared with 92% for Configuration A. This suggests that object-region extraction may occasionally discard spatial evidence when a query contains only a restricted portion of the original viewport. However, Configuration A remained substantially faster under this condition and achieved the best overall balance between recognition robustness and computational efficiency.

5 Discussion

5.1 Effects of Viewport-Aware Robust Feature Learning

The proposed method achieved an overall recognition rate of 97.5% across 18,000 transformed queries. This result indicates that transformation-consistent feature points can preserve content identity under both conventional video modifications and immersive-video-specific transformations. Representative-frame selection reduced temporal redundancy, while viewport-aware processing limited feature extraction to regions containing stable visual information.

The method maintained recognition rates of at least 95% under logo insertion, caption insertion, compression, codec conversion, resolution change, rotation, flipping, grayscale conversion, and photometric adjustment. In particular, grayscale, brightness, and contrast transformations produced recognition rates of 97%, suggesting that the selected feature points relied primarily on local structural patterns rather than absolute color or intensity values. Projection conversion and base-

viewpoint shift both achieved 100% recognition, demonstrating the effectiveness of viewport-based region processing against changes in immersive video representation.

Spatial cropping produced the greatest performance degradation. The recognition rates decreased to 92% for monoscopic cropping and 89% for stereoscopic cropping. Unlike global transformations, cropping permanently removes portions of the visual evidence used during content registration. Stereoscopic cropping further changes the visible regions of the two views, reducing the number of consistent correspondences available for matching. These results indicate that the remaining limitation is associated mainly with spatial information loss rather than signal-level degradation.

The preprocessing comparison further explains the performance improvement. Full-frame matching without preprocessing achieved only 59.3% recognition, whereas retaining the preprocessing procedures without object-region extraction increased the recognition rate to 92.0%. Applying the complete method improved this result to 97.5%. The results suggest that viewport-aware region processing suppresses features affected by projection distortion, while object-region extraction provides an additional improvement by retaining visually meaningful correspondences.

The complete method also reduced the average response time from 5,577 ms to 1,029 ms compared with full-frame matching. This reduction indicates that the proposed processing steps improve both robustness and efficiency by decreasing the number of irrelevant candidate features. However, object-region extraction reduced the recognition rate for monoscopic cropping from 95% to 92%, suggesting that restrictive region selection may discard useful evidence when only a small portion of the original viewport remains. An adaptive region-selection criterion may therefore improve robustness under severe spatial cropping.

5.2 Practical Applicability and Limitations

The proposed method can be applied to immersive content retrieval, duplicate detection, and unauthorized redistribution monitoring. Because content identity is determined from visual features rather than filenames or embedded metadata, the method can identify transformed copies even when container information has been removed or modified. The generated feature descriptors can also be associated with ownership and distribution metadata in a content database, supporting copyright management for 360-degree and immersive videos.

The average query response time of approximately 1.03 s demonstrates the feasibility of query-level content identification. This processing speed is suitable for repository inspection, platform upload verification, and post-distribution monitoring. However, the current results do not demonstrate continuous frame-by-frame recognition during immersive video playback. Integration with an OTT platform or an HMD-based application would require additional evaluation of end-to-end latency, network transmission, and database access time.

Several limitations should be considered. First, the experiment used 100 original immersive videos and 18,000 transformed queries. Although the transformation conditions covered signal, geometric, temporal, and immersive-video-specific modifications, the dataset may not represent every content genre, projection format, capture device, or distribution environment. Validation using additional public immersive video datasets would improve the generalizability of the findings.

Second, spatial cropping remained the most difficult transformation. Monoscopic and stereoscopic cropping produced recognition rates of 92% and 89%, respectively, because substantial portions of the registered visual evidence were removed. The monoscopic-cropping ablation result further showed that fixed object-region extraction can occasionally exclude features that remain useful in a restricted viewport. Adaptive region expansion or multi-region matching should therefore be considered for severely cropped queries.

Finally, the evaluation used a fixed database containing 100 reference videos and a high-performance computing environment. The measured response time does not directly represent performance in a large-scale commercial repository. Future evaluation should examine database scalability, memory consumption, indexing time, and retrieval latency as the number of registered videos increases. Comparisons with recent learning-based local feature and matching methods would also clarify the relative effectiveness of the proposed method under a common immersive-video benchmark.

6 Conclusion

This study proposed a viewport-aware content recognition method using transformation-robust feature points for immersive video. The method addresses the instability of conventional visual features caused by equirectangular projection distortion, viewport changes, video processing, and spatial information loss.

The proposed method selects representative frames, converts immersive frames into viewport regions, removes regions with insufficient visual information, and extracts object-centered feature points. Feature points that remain consistent between original and transformed content are retained as robust descriptors for reference indexing and query matching.

The method was evaluated using 100 original immersive videos and 18,000 queries generated under 18 transformation conditions. It achieved an overall recognition rate of 97.5%, a misrecognition rate of 2.0%, and a non-recognition rate of 0.5%. The average response time was approximately 1.03 s. The complete method also increased the recognition rate by 38.2 percentage points and reduced latency by 81.6% compared with full-frame matching without preprocessing.

The results demonstrate that viewport-aware processing and transformation-consistent feature selection can improve the robustness and efficiency of immersive video identification. The method can support content retrieval, duplicate detection, and unauthorized redistribution monitoring without relying on filenames or embedded metadata. Future work will focus on adaptive region selection for

severely cropped content, validation using larger public datasets, comparison with recent learning-based feature-matching methods, and scalability evaluation for large reference databases.

References

Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110(3):346–359. https://doi.org/10.1016/j.cviu.2007.09.014

Coors B, Condurache AP, Geiger A (2018) SphereNet: learning spherical representations for detection and classification in omnidirectional images. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds) Computer Vision–ECCV 2018. Lecture Notes in Computer Science, vol 11213. Springer, Cham, pp 525–541. https://doi.org/10.1007/978-3-030-01240-3_32

Corbillon X, Simon G, Devlic A, Chakareski J (2017) Viewport-adaptive navigable 360-degree video delivery. In: Proceedings of the IEEE International Conference on Communications, pp 1–7. https://doi.org/10.1109/ICC.2017.7996611

Cruz-Mota J, Bogdanova I, Paquier B, Bierlaire M, Thiran JP (2012) Scale invariant feature transform on the sphere: theory and applications. Int J Comput Vis 98(2):217–241. https://doi.org/10.1007/s11263-011-0505-4

DeTone D, Malisiewicz T, Rabinovich A (2018) SuperPoint: self-supervised interest point detection and description. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 224–236. https://doi.org/10.1109/CVPRW.2018.00060

He S, Yang X, Jiang C, Liang G, Zhang W, Pan T, Wang Q, Xu F, Li C, Liu J, Xu H, Huang K, Cheng Y, Qian F, Zhang X, Yang L (2022) A large-scale comprehensive dataset and copy-overlap aware evaluation protocol for segment-level video copy detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 21054–21063. https://doi.org/10.1109/CVPR52688.2022.02041

Jiang YG, Jiang Y, Wang J (2014) VCDB: a large-scale database for partial copy detection in videos. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision–ECCV 2014. Lecture Notes in Computer Science, vol 8692. Springer, Cham, pp 357–371. https://doi.org/10.1007/978-3-319-10593-2_24

Jeong M, Nang J (2021) A fusion of CNN-based frame vector for segment-level video partial copy detection (in Korean). J KIISE 48(1):43–50. https://doi.org/10.5626/JOK.2021.48.1.43

Kang IS, Seo YH, Kim DW (2017) Blind digital watermarking methods for omnidirectional panorama images using feature points (in Korean). J Broadcast Eng 22(6):785–799. https://doi.org/10.5909/JBE.2017.22.6.785

Kang S, Jeong M, Nang J (2023) A generation method of segment-level fingerprint-based transformer for video partial copy detection (in Korean). J KIISE 50(3):257–262. https://doi.org/10.5626/JOK.2023.50.3.257

Kim AY, An E, Seo KD (2019) An advanced QER selection algorithm based on MMT protocol for 360-degree VR video streaming (in Korean). J Broadcast Eng 24(6):948–955. https://doi.org/10.5909/JBE.2019.24.6.958

Kim KB (2013) Panoramic image reconstruction using SURF algorithm (in Korean). J Korea Soc Comput Inf 18(4):13–18. https://doi.org/10.9708/jksci.2013.18.4.013

Lee GW, Han JK (2017) Luminance compensation using feature points and histogram for VR video sequence (in Korean). J Broadcast Eng 22(6):808–816. https://doi.org/10.5909/JBE.2017.22.6.808

Lee SB, Nang J (2019) A near-duplicate image detection system for design contents using SIFT (in Korean). KIISE Trans Comput Pract 25(5):257–262. https://doi.org/10.5626/KTCP.2019.25.5.257

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Moon WJ, Seo YH, Kim DW (2019) SIFT feature-based digital watermarking method for VR images (in Korean). J Broadcast Eng 24(6):1152–1162. https://doi.org/10.5909/JBE.2019.24.6.1162

Park K, Kim W, Lee R, Jang RY, Choi MS (2022) Key frame detection using contrastive learning (in Korean). J Broadcast Eng 27(6):897–905. https://doi.org/10.5909/JBE.2022.27.6.897

Potje G, Cadar F, Araujo A, Martins R, Nascimento ER (2024) XFeat: accelerated features for lightweight image matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2682–2691. https://doi.org/10.1109/CVPR52733.2024.00259

Revaud J, Weinzaepfel P, de Souza CR, Pion N, Csurka G, Cabon Y, Humenberger M (2019) R2D2: repeatable and reliable detector and descriptor. In: Advances in Neural Information Processing Systems, vol 32, pp 12405–12415

Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: an efficient alternative to SIFT or SURF. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

Son HJ, Han JK (2018) Advanced seam finding algorithm for stitching of 360 VR images (in Korean). J Broadcast Eng 23(5):656–668. https://doi.org/10.5909/JBE.2018.23.5.656

Su YC, Grauman K (2017) Learning spherical convolution for fast features from 360° imagery. In: Advances in Neural Information Processing Systems, vol 30, pp 529–539

Tyszkiewicz MJ, Fua P, Trulls E (2020) DISK: learning local features with policy gradient. In: Advances in Neural Information Processing Systems, vol 33, pp 14254–14265

Zhao Q, Feng W, Wan L, Zhang J (2015) SPHORB: a fast and robust binary feature on the sphere. Int J Comput Vis 113(2):143–159. https://doi.org/10.1007/s11263-014-0787-4

Zhao X, Wu X, Chen W, Chen PCY, Xu Q, Li Z (2023) ALIKED: a lighter keypoint and descriptor extraction network via deformable transformation. IEEE Trans Instrum Meas 72:1–16. https://doi.org/10.1109/TIM.2023.3271000