EVALUATION FRAMEWORK FOR 360-DEGREE VISUAL CONTENT COMPRESSION WITH USER VIEW-DEPENDENT TRANSMISSION

Navid MAHMOUDIAN BIDGOLI, Thomas MAUGEY, Aline ROUMY

INRIA Rennes Bretagne-Atlantique

ABSTRACT

Immersive visual experience can be obtained by allowing the user to navigate in a 360-degree visual content. These contents are stored in high resolution and need a lot of space on the server to store them. The transmission depends on the user's request and only the spatial region which is requested by the user is transmitted to avoid wasting network bandwidth. Therefore, storage and transmission rates are both critical. Splitting the rates into storage and transmission has not been formally considered in the literature for evaluating 360-degree content compression algorithms. In this paper, we propose a framework to evaluate the coding efficiency of 360-degree content while discriminating between storage and transmission rate and taking into account user dependency. This brings the flexibility to compare different coding methods based on the storage capacity on the server and network bandwidth of users.

***Index Terms*—** 360-degree content, omnidirectional, Evaluation framework, Compression, User-dependent transmission

1. INTRODUCTION

With spherical visual content, also known as 360-degree or omnidirectional images and videos, users can freely observe the surrounding area and interact with the scene by means of head-mounted displays (HMD) or hand-held pointing devices as with the Google Maps Platform. These contents are compressed and stored on a server. Then, upon request of a direction by a user, only a small portion of the entire 360-degree content is displayed. Another characteristic of these contents is that each requested content is displayed with high resolution to provide a realistic immersive experience for the user. Hence these contents must be compressed efficiently in a way that the compressed stream can adapt to the user's request. This introduces naturally three criteria: storage S (the amount of data stored on the server), transmission rate R (the amount of data sent to the user) and the distortion D (quality of the view displayed to the user).

The goal of this paper is to propose a framework to evaluate the efficiency of 360-degree content compression algorithms that takes into account the three above mentioned criteria. Despite several contributions that have been made to improve user-dependent compression, there is no proper evaluation framework to compare different coding schemes. This might be explained by the fact that the proposed 360-degree content rely on classical 2D compression standards. Indeed, compressing 360-degree content generally first consists in projecting the spherical content to a new 2D representation before coding the data. Different projections are used, e.g., equirectangular [1], cube map [2, 3], dodecahedron [4], etc. A first approach to transmit the data to the user consists in compressing and sending the projected content [5, 6]. A second approach avoids sending the whole content and the projection is further mapped into several entities to adapt to the user's requested direction. For instance, [7, 8] partition the equirectangular image into several tiles. Each tile is encoded independently and during transmission, the tiles related to the requested area are transmitted. In pyramid projection, the spherical content is mapped to multiple pyramids where the base of each pyramid presents a pre-defined viewport in higher resolution and the rest of the pyramid represents the non-viewport part in lower resolution. Upon user request, the pyramid whose base is closer to the user's viewing direction [2] will be transmitted to the user.

Since the existing 360-degree content compression algorithms are based on 2D compression algorithms, the evaluation of these coding schemes still relies on classical 2D metrics. In [5] the classical rate-distortion (RD) plots are used for the evaluation because the whole content is sent to the user ($R = S$). The novelty in [5] is the computation of the average users' viewport quality, which is approximated by a weighted spherical Peak Signal-to-Noise Ratio (PSNR). When the whole content is sent, the authors in [6] provide a subjective quality evaluation and compare the results with a range of objective quality metrics. In [9], a rate allocation strategy is proposed to lower the transmission rate, therefore only R is considered and not the trade-off between R and D. In [10], RD curves and sum of storage are used for the evaluation and the trade-off between S and D is not considered. To reduce the computational cost, they approximate the distortion using a set of discrete viewing orientations that does not depend on the users' navigation.

To the best of our knowledge, none of the existing evaluation methods consider jointly storage, transmission rate and distortion. In this paper, we propose a user-dependent evaluation framework to compare different 360-degree content compression methods in which the distortion at the user side is considered by computing the coding error in the viewport shown to users and at the same time, it differentiates the transmission rate from storage. For that, first, we propose to use Bjontegaard metric [11] to compute average storage and transmission rate saving for the same quality experienced by the user. Second, we propose iso points to compare methods when the system must satisfy some constraints. Finally, we extend Bjontegaard metric by combining the transmission rate and storage to take into account further constraints in the evaluation. The evaluation framework brings the ability to achieve a compromise between the capacity of the server to store the data and users' bandwidth and to have a clear understanding on the benefit of one method with respect to (w.r.t.) another one.

The rest of the paper is organized as follows. In Section 2 the 360-degree content compression with user-dependent transmission is formulated. Section 3 explains the proposed evaluation framework. We illustrate the evaluation methodology on existing coding schemes in Section 4.

This work was partially supported by the Cominlabs excellence laboratory with funding from the French National Research Agency (ANR-10-LABX-07-01) and by the Brittany Region (Grant No. ARED 9582 InterCOR).

2. COMPRESSION WITH USER-DEPENDENT TRANSMISSION

Available coding schemes are mostly able to compress 2D image/video formats. Therefore, the spherical image/video Γ is first mapped to one or several rectangular planes and then these mapped representations are compressed with existing 2D coding schemes. To avoid suboptimal usage of the network bandwidth, the mapping also enables to extract and transmit the spatial region requested by the user's direction θ. If we denote the mapping function as $m : \Gamma \mapsto I$, where I is the corresponding 2D representation, the encoding procedure can be considered as

$$(f \circ m)(\Gamma) = (b_1, \ldots, b_n)$$

where f is the encoding function and $b_i, i = 1, .., n$ are independently extractable bitstreams which are stored on the server. The storage size on the server is

$$S = |f \circ m(\Gamma)| = \sum_{i=1}^{n} |b_i| \tag{1}$$

A head rotation θ of the user triggers a request of a subpart $\gamma(\theta)$ of the spherical image Γ. Then, the server extracts parts of the stored bitstreams $i \in \mathcal{I}_\theta$ related to $\gamma(\theta)$ and sends them to the user for decoding. The transmission rate of request θ which is equal to the size of extracted bitstreams in $\mathcal{I}_\theta$ can be written as

$$R_\theta = \sum_{i \in \mathcal{I}(\theta)} |b_i| \tag{2}$$

where $R_\theta \leq S$. After receiving the requested bitstreams, the decoder h_θ decodes the requested region and maps it back on the sphere:

$$m^{-1} \circ h_\theta : (b_i)_{i \in \mathcal{I}_\theta} \rightarrow \hat{\gamma}(\theta).$$

Note that due to lossy compression $\gamma(\theta) \neq \hat{\gamma}(\theta)$. To measure a realistic quality of experience at the user side, the distortion D_θ is computed in the viewport $v(\theta)$ shown to the user which is the image plane tangential to the sphere at a point defined by the radius vector with rotation θ (w.r.t. the sphere coordinate system)

$$D_\theta = ||v(\theta) - \hat{v}(\theta)||_2^2.$$

Both distortion D_θ and rate R_θ vary significantly with the direction θ. Therefore, to be able to compare performance between different methods, the expected values over all users' requests are computed for these quantities. Compared to classical 2D video/image coding, the use of expected values over all users' requests for the transmission rate and distortion is new and comes from the user-dependent navigation. A method which has lower expected distortion $\mathbb{E}_\theta(D_\theta)$ for the same amount of storage S and same expected transmission rates $\mathbb{E}_\theta(R_\theta)$ is preferred. Therefore, unlike conventional 2D image/video evaluation methodologies where 2D rate-distortion (RD) plots are used for comparison, here we need to compare them with 3D Storage-Rate-Distortion (S-R-D) curves (for simplicity from now on we omit the term *expected* and simply call S-R-D). By changing the Quantization Parameter (QP) of the encoder, different qualities of the 360-degree content are generated. It is worth noting that since the only free parameter that is adjustable is the QP value, the S-R-D values generate 3D curves.

3. PROPOSED EVALUATION FRAMEWORK

Performing a formal comparison between 3D curves is a difficult task especially when the curves have different domains as is the case for typical S-R-D curves, see Fig. 1. We propose 3 ways to compare methods based on the constraints imposed by the system.

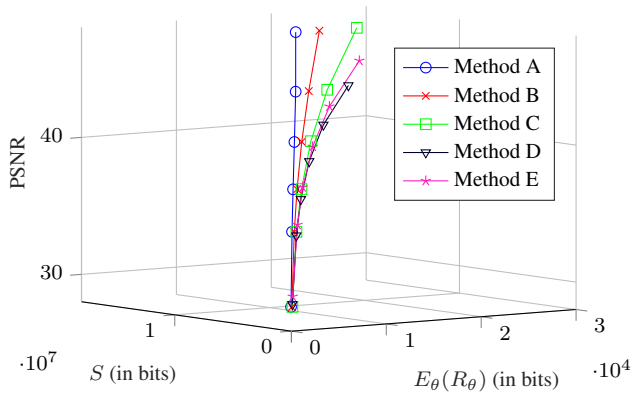

Fig. 1: Example of S-R-D curve. Comparison between different methods is difficult in 3D. We propose 3 solutions to compare between methods.

3.1. Average storage and transmission rate saving

Bjontegaard-Delta (BD) rate [11] is a commonly used metric as it performs a fair comparison between two coding methods. More precisely, a common PSNR range is determined and the average bitrate reduction/increase is computed for this PSNR range. We extend the BD measure for storage and transmission rate by projecting S-R-D curves to R-D and S-D planes and we name them BD-R and BD-S respectively. The BD-S represents the average reduction/increase in the amount of data stored on the server for the same PSNR range omitting the transmission rate. BD-R is equivalent to classical BD-R and represents the average bit rate saving/increase for the same PSNR which is transmitted to each user and storage is not taken into account. To evaluate different methods, the pair (BD-R, BD-S) must be considered jointly.

It is worth noting that in [10], the BD-R is computed and the sum of storage over all stored qualities are compared. However the sum of stored qualities may not cover the same range of distortion, which leads to an unfair comparison. Introducing BD-S solves this issue because it considers the same range of PSNR for both methods.

Two problems can happen with this metric. The first one is due to the fact that sometimes it is not clear how to compare between pairs. For instance, assume (-20%, 10%) and (15%, -10%) for (BD-R, BD-S) pair between 2 methods (negative values represent percentage saving w.r.t. the reference). Here the former performs better on average in terms of transmission rate and the latter consumes less space on the server for the same PSNR. This problem will be tackled in Section 3.3. Second problem is that BD represents the average performance, but sometimes there are some constraints that the scheme must satisfy. Therefore, we propose iso points in the next subsection to evaluate methods not based on their average behavior, but rather on their performance under some constraint.

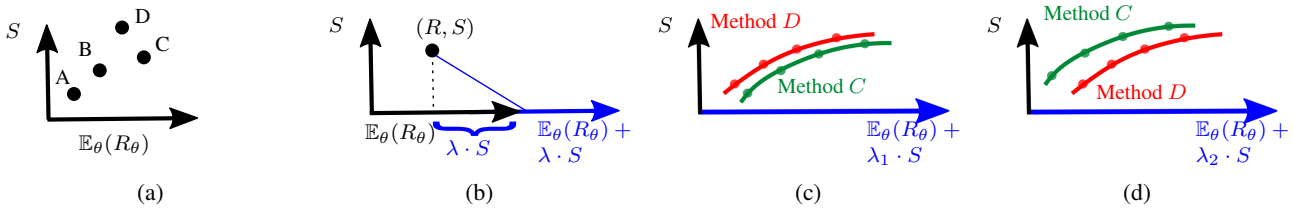

Fig. 2: New metric for the analysis of immersive coding schemes. (a) iso distortion points: each point corresponds to the R,S performance of a compression method for a fixed distortion. Therefore, all points have the same distortion. (b) Projection of the SR plane onto the new axis $R + \lambda S$ (in blue). (c) and (d) Projected S-R-D curves for different λ where $\lambda_1 < \lambda_2$.

3.2. Iso points

In this section, we propose a set of evaluation tools that can be used when the system must satisfy some constraints. For instance, the service might be constrained to serve the clients with a minimum distortion level d. To choose and compare the coding schemes, we introduce the iso-distorsion points:

$$\{(S_i, R_i, D_i) | D_i = d \text{ and } i = \text{indices of method}\}$$

where linear interpolation of the S-R-D curve is used to compute the isocurve.

Similarly, iso storage (iso transmission rate) values show that at the same storage (transmission rate) what are the distortion-transmission rate (distortion-storage) values. Depending on what are the constraints of the streaming system (distorion or transmission rate or storage), one can plot iso points and compare the methods on iso values domain.

An example of iso distortion values is shown in Fig. 2a. Comparing method A and B which have iso distortion values, reveal that method A consumes less bandwidth than method B and it needs lower space on the server to store data. Therefore, method A is preferred over method B. Sometimes isocurves do not help to determine which method is better than the other. For instance, in Fig. 2a method C takes lower space on the server and method D performs better in terms of transmission rate.

3.3. Weighted BD

To solve the ambiguities occurred with isocurves and (BD-R,BD-S) pairs, there is a need to adjust the tradeoff between rate and storage. This can be done by introducing a weighted combination:

$$\mathbb{E}_\theta(R_\theta) + \lambda \cdot S \tag{3}$$

where λ balances the importance between storage and transmission rate and is determined by industrial constraints (a high value of λ means that storage is more important than transmission rate). With this definition (3), a point in the R-S plane is mapped onto a point on the axis depicted in blue in Fig. 2b. This mapping also yields a projection of the 3D S-R-D curve onto the 2D plane defined by the distortion D and the weighted combination (3). Examples of such curves are shown in Fig. 2c. Then, from two such curves, a BD measure can be computed. This is called weighted BD.

The choice of the regularizer parameter λ is crucial, and may yield opposite conclusions when comparing compression methods, see Figs. 2c and 2d. For instance, (3) can result from

$$\alpha \cdot \mathbb{E}_\theta(R_\theta) + \beta \cdot S \tag{4}$$

where the regularizers α, β can be used to impose another criterion like delay time for each method. For example, if β represents the time per byte that it takes to read the data from the hard drive in the server and α represents the inverse of the bandwidth, then (4) represents the overall delay time which consists of delay time for the server to read data from its hard drive plus the delay time for the data to be received by the users through the network. By performing BD over this new metric we can compare the delay time over the same PSNR values between two methods.

4. EXPERIMENTAL ILLUSTRATION

In this section, we illustrate the usefulness of the proposed framework for immersive compression scheme analysis. Four large size 8960x4480 panoramic images from the SUN360 database [12] are considered: 2 from indoor environments and 2 from outdoor environments. Note that the evaluation framework can also be applied in the same manner to 360-degree videos. As for the compression methods, we consider tile-based coding of equirectangular images with 3 tile sizes (640x320, 1280x640 and 4480x2240) which partition the panoramic image into 14x14, 7x7 and 2x2 tiles respectively. We denote them by *E. 14x14*, *E. 7x7* and *E. 2x2* respectively in the following. Each tile is encoded independently using HEVC intra-coding [13]. Upon user request, the server sends only the tiles related to the requested viewport. If any tile has been sent already in a previous request, the server avoids sending the tile again. We also implemented the cube map representation with 2 different face resolutions: 2560x2560 and 3008x3008 (the sum of six cube faces of size 2560x2560 is almost equal to the panoramic image size). We denote them by *C. lower* and *C. higher* respectively. Classically in cube representation, the server sends the whole cube. Here, in order to be similar to the tile-based approach, each face is encoded separately with HEVC intra-coding. Then, the server sends only the faces of the cube which are related to the requested viewport.

Our proposed evaluation framework does not rely on a specific user navigation model: it simply needs a set of user head position no matter how they are generated. We restrict our experiments to images, but one can perform similarly with videos. Since there is no long duration navigation dataset for this large size image database, we created one by following the recommendations in [5]. In particular, it is observed in [5] that users more frequently view areas around the equator (up to latitude ± 30) than the poles. Moreover, when users are moving their head by changing the longitude and latitude angles, it is more likely for them to continue their previous direction rather than changing the head movement to the opposite direction. Therefore, we simulated user's navigation by defining three probabilities p_1, p_2, p_3, for each of the longitude and latitude angles of the user's head orientation. p_1 is the probability of continuing the previous head movement, p_2 is the probability of staying at the current angle and $p_3, p_1 > p_3$ is the probability of changing movement to the

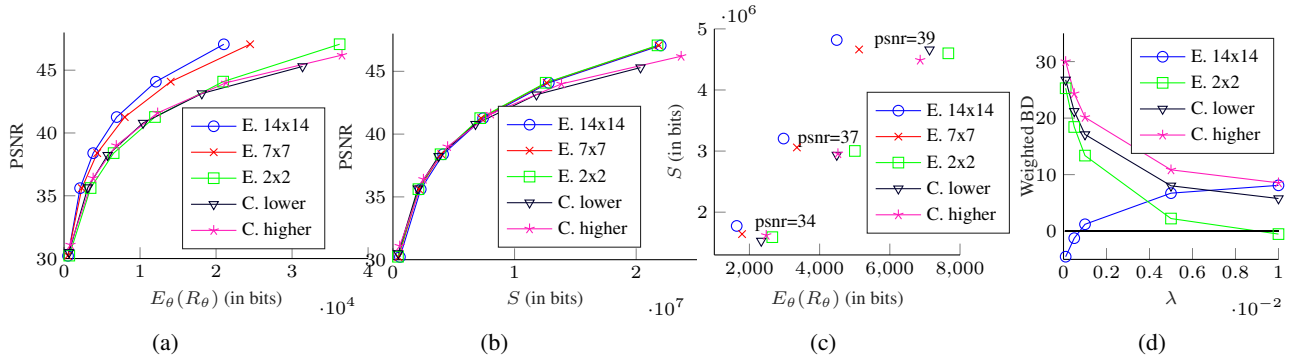

Fig. 3: Curves for user-dependent compression and transmission of image pano_aagymbthhtcgfb in SUN360 dataset. (a) R-D curve. (b) S-D curve. (c) Iso distortion points. (d) weighted BD for different λ.

opposite direction. The latitude is initialized randomly in range ± 30. We generate navigation for 20 users where each has 600 requests (in total 12000 requests per image). For each image $\mathbb{E}_\theta(R_\theta), \mathbb{E}_\theta(D_\theta)$ are computed by averaging over all users' requests.

4.1. Storage and transmission rate saving

Intuitively, approaches with smaller tile size should perform better in terms of transmission, but be less efficient in terms of storage. This is because small tiles brings more flexibility to send only the parts which are requested, but larger tiles result in better encoding performance. The proposed (BD-R, BD-S) criterion allows to assess quantitatively to what extent the tile size influences the tradeoff between transmission rate and storage, as shown in Table 1. For instance, the first two rows of Table 1 give the (BD-R, BD-S) pair averaged over all images, where 7x7 tiling is used as a reference (positive values show they perform worse than 7x7 tiling and negative values show they perform better). Interestingly, we observe that the tile size influences much more the transmission rate (from $\sim$-6% to $\sim$32%) than the storage ($\sim$-4% to $\sim$10%).

Our proposed S-R-D characterization of the performance of an immersive coding scheme can also be plotted in terms of joint R-D and S-D curves, as in Figs. 3a and 3b (for one image). In this representation, we observe that cube map approach saturates at high bit rates in both R-D and S-D curves. This can be explained by the fact that the original images are stored in equirectangular format, and there is a loss in quality in the cubemap approach, due to the extra projection from the equirectangular to the cube map representation.

4.2. Iso distortion values

The proposed iso distortion points allow to compare schemes at a given operating point and can help guiding the design of the compression scheme. For instance, one can evaluate the influence of the face resolution on the performance of the cubemap approach in Fig. 3c. At low PSNR (34 dB), lower face resolution achieves a smaller transmission rate and storage than higher face resolution, whereas, at high PSNR (39 dB), higher face resolution performs better in both storage and transmission rate. This highlights the strategy that for cube map projection, it is better to store cube faces with lower resolution at lower bitrates (low PSNRs) and at high bitrates cube faces resolution must be increased.

4.3. Weighted BD

Weighted BD was introduced in this paper to deal with cases where the comparison between two coding schemes is not straightforward.

Table 1: BD measures averaged over all images. 7x7 tiling is used as a reference. The first 2 rows are (BD-R, BD-S) pair. The last 3 rows are weighted BD computed over the same PSNR range for (3).

	E. 14x14	E. 2x2	C. lower	C. higher
BD-R	-5.67 %	27.71 %	28.74 %	32.03 %
BD-S	9.84 %	-3.94 %	2.97 %	5.68 %
$\lambda = 0.01$	8.10 %	-0.52 %	5.77 %	8.55 %
$\lambda = 1e^{-3}$	1.21 %	13.37 %	17.10 %	20.14 %
$\lambda = 1e^{-4}$	- 4.52 %	25.28 %	26.77 %	30.02 %

This occurs in particular, when one scheme achieves a lower transmission rate but a higher storage. By contrast, the weighted-BD (3) can help to discriminate between the schemes, by fixing λ, which is interpreted as the cost between storage and transmission rate, as shown in the last 3 rows of Table 1. For example, in our experiment the storage is about 700-1000 times larger than the transmission rate. Imposing $\lambda = 1e^{-3}$ means that storage and transmission rate are both of the same importance. In this scenario, 7x7 tiling performs better than the other ones. Using smaller $\lambda = 1e^{-4}$, transmission is getting more important and 14x14 tiling should be chosen. Weighted-BD for different λs are plotted in Fig. 3d, where the lowest wighted-BD corresponds to the best method. More precisely, when $\lambda < 0.0008$, 14x14 tiling achieves the best performance, whereas the reference 7x7 tiling is better in $0.0008 \leq \lambda < 0.009$ and 2x2 tiling should be chosen for $\lambda \geq 0.009$.

5. CONCLUSION

In this paper, we proposed an evaluation framework to compare 360-degree content compression schemes for user-dependent transmission. We considered both storage and transmission rate jointly as they both affect decision making depending on the constraints we have in the streaming system. For tiling the equirectangular image, we showed that tiling the image into around 7x7 tiles, brings a good compromise between storage and transmission rate when storage and transmission rates are both of the same importance. Using iso distortion curves, we showed that for cube map we should consider a multi-resolution strategy where at lower bitrates (low PSNR values) cube faces with lower face resolution perform better in terms of both storage and transmission rate. At high bitrates (higher resolution) by increasing the cube faces resolution we can expect better performance again in both transmission rate and storage.

6. REFERENCES

[1] J.P. Snyder, *Flattening the Earth: Two Thousand Years of Map Projections*, University of Chicago Press, 1993.

[2] E. Kuzyakov and D. Pio, "Next-generation video encoding techniques for 360 video and vr," 2016, [Online]. Available: https://code.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/.

[3] King-To Ng, Shing-Chow Chan, and Heung-Yeung Shum, "Data compression and transmission aspects of panoramic videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 82–95, Jan 2005.

[4] C. Fu, L. Wan, T. Wong, and C. Leung, "The rhombic dodecahedron map: An efficient scheme for encoding panoramic video," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 634–644, June 2009.

[5] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *2015 IEEE International Symposium on Mixed and Augmented Reality*, Sep. 2015, pp. 31–36.

[6] E. Upenik, M. Rerabek, and T. Ebrahimi, "On the performance of objective metrics for omnidirectional visual content," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2017, pp. 1–6.

[7] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016, pp. 601–605.

[8] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 1–6.

[9] M. Hosseini and V. Swaminathan, "Adaptive 360 vr video streaming: Divide and conquer," in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 107–110.

[10] A. Zare, A. Aminlou, and M. M. Hannuksela, "Virtual reality content streaming: Viewport-dependent projection and tile-based techniques," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 1432–1436.

[11] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *VCEG-M33*, 2001.

[12] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 2695–2702.

[13] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.