A FRAMEWORK FOR SURFACE LIGHT FIELD COMPRESSION

Xiang Zhang[⋆†], *Philip A. Chou*[§], *Ming-Ting Sun*[†], *Maolong Tang*[†], *Shanshe Wang*[⋆], *Siwei Ma*[⋆], *Wen Gao*[⋆]

[⋆] Institute of Digital Media, Peking University, Beijing, China
[†] Department of Electrical Engineering, University of Washington, Seattle, WA, USA
[§] 8i Labs, Inc., Seattle, WA, USA

ABSTRACT

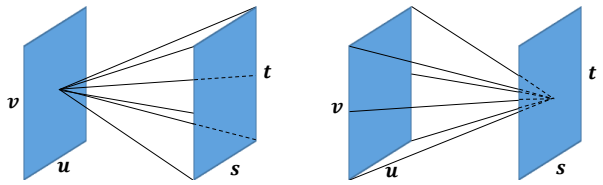
Surface Light Fields (SLF) have previously been proposed for representing 3D scenes under complex lighting conditions, enabling immersive viewing experiences from arbitrary observation directions. In this work, we present a new approach for SLF representation and a framework for SLF compression. Specifically, the SLF is compactly represented in a B-Spline wavelet basis. This representation is capable of modeling diverse surface materials and complex lighting conditions. The coefficients of the B-Spline wavelet are then compressed by removing the spatial redundancy over surface points. Compared with image based light field compression, the proposed scheme is functionally advanced because it enables rendering objects from arbitrary viewpoints with both good quality and high efficiency. In terms of bitrate and distortion, experimental results have shown that the proposed method can achieve competitive performance but with much lower decoder computational complexity, indicating its potential in practical virtual and augmented reality applications.

Index Terms— Surface light field, VR, AR.

1. INTRODUCTION

In emerging virtual reality (VR) and augmented reality (AR) applications, it is important to be able to render a scene from arbitrary points of view, allowing free-viewpoint navigation for example. While conventional computer graphics (CG) allow synthesis of CG-modeled scenes from arbitrary points of view, the photorealism of natural scenes using CG models is elusive, at least without extreme computation, especially in the presence of complex material and lighting phenomena.

Light Fields (LF) aim to provide photo-realistic renderings of 3D scenes from a range of viewpoints even in the presence of such complex material and lighting phenomena. An LF is most frequently represented as a 4D function of a light ray [1]. There are two types of such an LF representation, as shown in Fig. 1. A sub-aperture representation is essentially a collection of images captured from different viewpoints, while a lenslet representation is essentially a collection of images (here called *view maps*) of the color of each point on a plane as the point appears from different directions.



(a) Sub-aperture representation (b) Lenslet representation

Fig. 1. Two-plane light field representations [1], where (u, v) and (s, t) indicate the camera plane and the focal plane.

The LF representation requires a large amount of information, hindering transmission, storage, and application development. Therefore, LF compression has attracted extensive attention recently, including standardization efforts in JPEG and MPEG. Almost all of the recent LF compression methods target LF images captured with a micro-lens array [2] or with a dense camera array. Some of them compress the LF representation as a natural image by exploiting intra-image similarities [3, 4, 5, 6]. Alternatively, by reordering sub-aperture images as a video, video coding techniques have been adopted to remove inter-image redundancies [7, 8, 9, 10, 11, 12, 13].

However, these LF methods make at most limited use of geometric information, and as a result have significant limitations. For one, since the images are captured by cameras at a discrete set of positions, view interpolation is necessary, requiring a dense camera array to avoid occlusion artifacts. Even more significantly, extrapolation of views outside a narrow range of view angles close to the original camera positions is generally not feasible. This all but eliminates conventional LF approaches for VR and AR applications in which arbitrary points of view of dynamic scenes must be generated.

A more efficient and flexible representation is the Surface Light Field (SLF) [14, 15, 16]. The SLF enables synthesis from an arbitrary viewpoint, interactive rendering, and rudi-

This work was supported in part by National Natural Science Foundation of China (61571017, 61632001), National Postdoctoral Program for Innovative Talents (BX201600006), Top-Notch Young Talents Program of China, High-performance Computing Platform of Peking University and China Scholarship Council (CSC), which are gratefully acknowledged.

mentary editing of the LF. Essentially, the SLF defines the light rays emanating from a point of the 3D scene. The SLF can be regarded as a function $f(\boldsymbol{\omega}|\boldsymbol{p})$, where $\boldsymbol{p}$ is the location in 3D of a surface point, and $\boldsymbol{\omega}$ is the direction of a ray emanating from the point. The SLF can be viewed as a generalization of the lenslet representation, if one considers the surface point at $\boldsymbol{p}$ as a point on the (s,t) plane. Since a SLF is a generalization of a LF, it can represent anything that a LF can represent. Moreover, it has the potential to be a more efficient representation. Indeed, for Lambertian or near-Lambertian objects, the view map at each point $\boldsymbol{p}$ is a constant or near-constant image, reducing $f(\boldsymbol{\omega}|\boldsymbol{p})$ essentially to a function only of $\boldsymbol{p}$, like a 2D CG texture map. An alternative view of an SLF is as a CG texture map whose value at every point is an image, which can be arbitrarily complex yet is frequently near-constant. Thus an SLF can also be viewed as a generalization of a CG texture map. In a sense, SLFs combine the best of LFs and CG modeling, allowing photo-realistic rendering from arbitrary points of view.

In this work, we propose a new SLF representation and method for its compression. In a nutshell, we propose to represent and compress the 4D SLF function $f(\boldsymbol{\omega}, \boldsymbol{p})$ as a separable linear transform $F(i,j)$, where i is an image frequency index and j is a spatial frequency index. Specifically, first, for every surface point $\boldsymbol{p}$, we use a linear transform to transform the view map $f(\boldsymbol{\omega}|\boldsymbol{p})$ into a sequence of image transform coefficients $\alpha_0(\boldsymbol{p}), \alpha_1(\boldsymbol{p}), \ldots$. Second, for every image transform coefficient $\alpha_i(\boldsymbol{p})$, we use a spatial transform (i.e., a transform across the surface) to transform $\alpha_i(\boldsymbol{p})$ as a function of $\boldsymbol{p}$ into a sequence of spatial transform coefficients.

2. SURFACE LIGHT FIELD REPRESENTATION BY B-SPLINE WAVELETS

We aim to determine an SLF representation that is efficient, robust, and scalable. Efficiency means that the representation is compact and friendly to compression. Robustness means that the representation is capable of approximating views from various directions of various surface materials under various lighting conditions. Scalability means that the representation can approximate simple (near-Lambertian surfaces) through arbitrarily complex view maps (reflective surfaces) using a bitrate commensurate with its complexity.

To this end, we propose to approximate the view map at each point $\boldsymbol{p}$ as a linear combination of basis functions,

$$f(\boldsymbol{\omega}|\boldsymbol{p}) \approx \sum_{i=0}^{N-1} \alpha_i(\boldsymbol{p}) \cdot g_i(\boldsymbol{\omega}), \tag{1}$$

where $g_i(\boldsymbol{\omega})$ is the i^{th} basis function and $\alpha_i(\boldsymbol{p})$ is the corresponding coefficient at point $\boldsymbol{p}$. N is the number of bases.

But the number of cameras to obtain the SLF is always limited in practical applications. As shown in Fig. 2, the

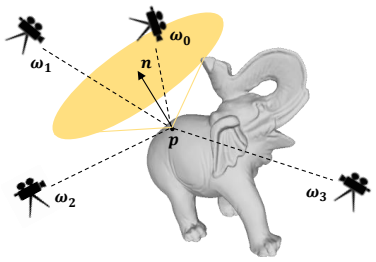


Fig. 2. Capturing a SLF by a number of cameras.

SLF of the object is captured by several cameras from different viewpoints. For a surface point $\boldsymbol{p}$, we denote its pixel values from different camera directions as a vector $\boldsymbol{c} = \{c_0, c_1, \ldots, c_{M-1}\}$, where M is the number of the observations. We also eliminate invalid observations caused by occlusions or out of camera view field. The corresponding camera directions $\boldsymbol{\omega}_m$ are parameterized by spherical coordinates with azimuth $\theta \in [-\pi, \pi]$ and elevation $\phi \in [-\pi/2, \pi/2]$. Further, we re-parameterize ϕ as $\gamma = \sin(\phi) \in [-1, 1]$, which ensures that equal areas in the (θ, γ) plane are equal areas on the sphere. In turn this ensures that any basis that is orthogonal on the (θ, γ) plane is orthogonal on the sphere.

Since we have only a limited number of cameras to measure the SLF, the number of valid observations M could be smaller than the number of basis functions N, making the problem underdetermined. Thus we regularize the solution. To be precise, let each element in matrix $\boldsymbol{G} \in \mathbb{R}^{M \times N}$ be $\boldsymbol{G}_{i,j} = g_i(\boldsymbol{\omega}_j)$. Then, given the observation vector $\boldsymbol{c}$ at point $\boldsymbol{p}$, we determine $\boldsymbol{\alpha}(\boldsymbol{p})$ as

$$\boldsymbol{\alpha}(\boldsymbol{p}) = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{c} - \boldsymbol{G}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_2^2, \tag{2}$$

where λ is a regularization factor. λ is significant for two reasons: 1) When the problem is underdetermined, i.e., $M < N$, λ avoids overfitting and yield compressible coefficients with reasonable range. 2) λ makes the solution robust to outliers due to camera noise and other imprecisions.

The design of basis functions is significant, since a good basis compacts the energy in the coefficients and make them easier to compress. In this work, we use the 2D separable B-Spline wavelets for the basis, because it is a good fit for the local variance characteristics of the SLF in realistic scenarios. It can be formulated as follows,

$$g_i(\theta, \gamma) = w_{i_0}\left(\frac{\theta}{2\pi}\right) w_{i_1}\left(\frac{\gamma}{2}\right), \tag{3}$$

where w_i is the i^{th} offset of the periodicized 1D B-Spline wavelet function

$$w_i(x) = \sum_{m \in \mathbb{Z}} \psi_o(2^s x - i + m2^s), \tag{4}$$

and ψ_o is the basic B-Spline wavelet with order o and scale s,

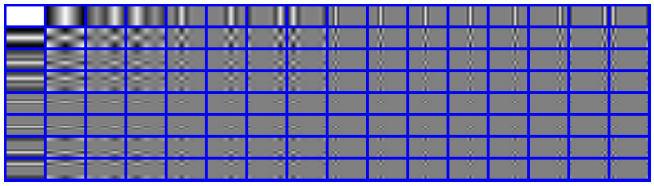


Fig. 3. B-Spline Wavelet functions for SLF representation.

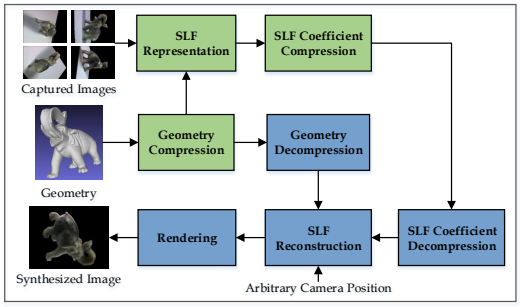


Fig. 4. The proposed SLF compression framework. Green and blue boxes indicate the processes on encoder and decoder sides, respectively.

which can be defined by the sum of cardinal B-Splines,

$$\begin{aligned} \psi_o(x) &= \sum_{n=0}^{3o-2} q_n N_o(2x-n), \\ q_n &= \frac{(-1)^n}{2^{o-1}} \sum_{j=0}^{o} \binom{m}{j} N_{2o}(n-j+1), \end{aligned} \tag{5}$$

where N_o is the cardinal B-Spline function. s_0 and s_1 are the scale of θ and γ, respectively, such that $i_0 = 0, 1, \ldots, 2^{s_0-1}$, $i_1 = 0, 1, \ldots, 2^{s_1-1}$, and $i = 0, 1, \ldots, 2^{s_0+s_1-1}$, indicating the total number of basis functions is $N = 2^{s_0+s_1}$.

We visualize the B-Spline wavelet basis functions in Fig. 3. There are 16 basis functions per row (the θ direction) and 8 basis functions per column (the γ direction). The top-left basis function is constant, so the corresponding coefficient is termed the DC coefficient as it represents the mean value. From the top-left to the bottom-right corner, the basis functions are able to describe more high-frequency signals.

3. SURFACE LIGHT FIELD COEFFICIENT COMPRESSION

Fig. 4 illustrates the pipeline of the proposed SLF compression. The input data include a point cloud that represents object geometry and a number of images captured from different points of view. The geometry of the point cloud, i.e., the 3D position of each point $\boldsymbol{p}$ is compressed as a Sparse Voxel Octree (SVO) [17, 18]. The SLF compression consists of the following substeps: 1) representation of the view map $f(\boldsymbol{\omega}|\boldsymbol{p})$ at each point $\boldsymbol{p}$ by a linear combination of B-Spline wavelet basis functions with coefficients $\boldsymbol{\alpha}(\boldsymbol{p})$ 2) independent compression of each wavelet coefficient $\alpha_i(\boldsymbol{p})$ by utilizing its spatial coherence across $\boldsymbol{p}$, and 3) decompression, reconstruction, and rendering of the SLF from arbitrary points of view.

To compress the SLF coefficients $\boldsymbol{\alpha}(\boldsymbol{p})$ for each point $\boldsymbol{p}$ accross the surface, the key issue is how to remove the spatial redundancy between neighboring points. We propose to apply the Region Adaptive Hierarchical Transform (RAHT) coding [17]. Since the distribution of points on a point cloud can be rather sparse compared to the whole space, RAHT adaptively applies the 2-point Haar transform to two spatially neighboring points and progressively groups them. After transformation, scalar quantization is applied to each coefficient given a quantization step size Q,

$$\hat{\boldsymbol{F}}_i = Round\left(\frac{\boldsymbol{F}_i}{Q}\right) Q, \tag{6}$$

where $\boldsymbol{F}_i$ denotes the coefficients transformed by RAHT and $\hat{\boldsymbol{F}}_i$ the quantized transformed coefficients. Then, the quantized coefficients $\hat{\boldsymbol{F}}_i$ are entropy encoded. On the decoder side, the coefficients can be recovered by entropy decoding, inverse quantization, and inverse RAHT.

We denote the recovered SLF coefficients as $\hat{\boldsymbol{\alpha}}$. Given $\hat{\boldsymbol{\alpha}}$, the SLF can be easily reconstructed at the decoder side as

$$\hat{f}(\boldsymbol{\omega}_v|\boldsymbol{p}) = \sum_{i=0}^{N-1} \hat{\alpha}_i(\boldsymbol{p}) \cdot g_i(\boldsymbol{\omega}_v), \tag{7}$$

where $\boldsymbol{\omega}_v$ is the direction of the virtual camera. Accordingly, free-view rendering can be achieved efficiently.

4. EXPERIMENTAL RESULTS

For evaluation, we use the dataset proposed in [15], containing two objects, *Elephant* and *Fish*. They have rich texture, specular surface and complex light illuminance. They are captured by 316 and 582 cameras, respectively, with 640×480 pixels. We use half of the cameras as input to represent the SLF and the other cameras as evaluation.

4.1. Reconstruction from Arbitrary Viewpoints

First, we evaluate the ability of the proposed scheme to reconstruct scenes from arbitrary viewpoints. This is one of the most significant advantages of the scheme over image-based or depth+image-based LF compression. Figs. 5 and 6 show reconstructions of *Elephant* and *Fish* from virtual viewpoints. It can be seen that the proposed scheme is adaptive and robust to different surface materials and light conditions.

4.2. Comparison with Image Based LF Compression

A fair comparison with image based LF compression is not straightforward, since the SLF representation is able to reconstruct viewpoints that are far from the original camera positions, while an image based LF cannot. However, we can

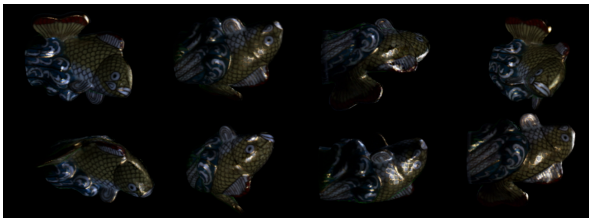


Fig. 5. Rendering *Fish* from arbitrary viewpoints.

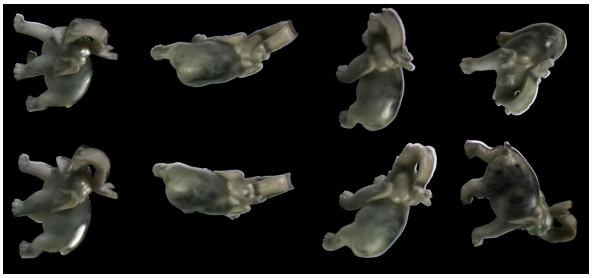


Fig. 6. Rendering *Elephant* from arbitrary viewpoints.

fairly compare our SLF compression scheme with an alternative scheme, which we call *Image-plus-Geometry Compression* (IGC), as illustrated in Fig. 7. At the encoder side, IGC compresses the captured images using a video codec. Like SLF compression, IGC also compresses the geometry using the same method. At the decoder side, instead of directly decoding the SLF representation and then rendering, IGC decodes the images, constructs the SLF representation from the decoded images, and finally uses the SLF representation to render arbitrary points of view. Thus the major difference is that the SLF representation is performed at the encoder in our scheme, and at the decoder in IGC. The latter increases the complexity of the decoder. The running time of decoder side between the proposed scheme and the IGC is compared in Tab. 1, where one can see that the proposed scheme is much faster than IGC. For encoder side, the complexity will increase because of the SLF representation. But considering practical applications, the decoder complexity is much more significant, and the codec can be further optimized.

Though IGC is not very practical, nonetheless we can compare the rate-distortion (RD) performance of IGC with

Table 1. Decoder running time comparison.

Running Time (s)	**IGC**	**Proposed**
Geometry Decompression	0.12	0.12
SLF Coef. Decompression	-	5.19
Images Decompression	1.92	-
SLF representation	35.13	-
Rendering	0.51	0.51
In Total	**37.68**	**5.82**

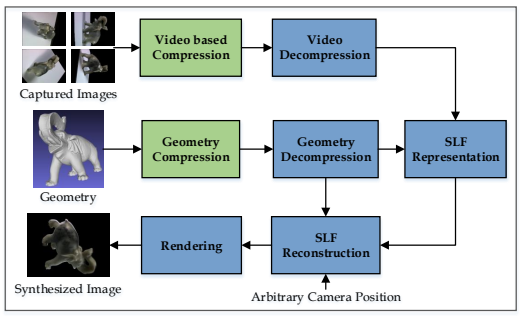


Fig. 7. The framework of IGC. Green and blue boxes indicate the processes on encoder and decoder sides, respectively.

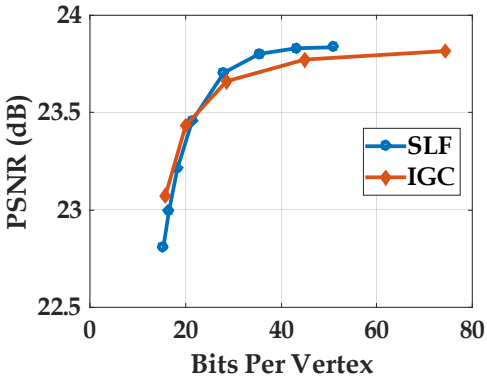


Fig. 8. RD performance comparisons between SLF and IGC.

that of the proposed SLF compression scheme to gain some insight. We compress the input images for IGC as a video sequence using the reference software HM-16.2 of High-Efficiency Video Coding (HEVC) [19]. Experimental results of the performance comparison are shown in Fig. 8, where one can see that the proposed SLF compression has RD performance competitive with or superior to IGC.

5. CONCLUSIONS

In this work, we propose a surface light field (SLF) compression framework for VR and AR applications. The advantage of SLF over image-based LF is that occlusions are more accurately modeled, thereby reducing the camera density needed for capture, and making the view maps easier to compress. We are able to achieve a compact and robust representation by approximating the view maps as a linear composition of B-Spline wavelets, and compressing the coefficients by removing the spatial redundancy over the surface. The proposed scheme is able to efficiently and robustly reconstruct images from a wide range of view directions. Experimental results indicate that the proposed method achieves competitive rate-distortion performance with low decoder complexity compared to an image-plus-geometry compression (IGC).

6. REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 31–42.

[2] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005.

[3] Y. Li, M. Sjostrom, R. Olsson, and U. Jennehag, "Coding of focused plenoptic contents by displacement intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1308–1319, 2016.

[4] ——, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 80–91, 2016.

[5] C. Conti, P. Nunes, and L. D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," in *IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW)*. IEEE, 2016, pp. 1–4.

[6] R. Monteiro, L. Lucas, C. Conti, P. Nunes, N. Rodrigues, S. Faria, C. Pagliari, E. da Silva, and L. Soares, "Light field HEVC-based image coding using locally linear embedding and self-similarity compensated prediction," in *IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW)*, 2016, pp. 1–4.

[7] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 338–343, 2000.

[8] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo-sequence-based 2-d hierarchical coding structure for light-field image compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1107–1119, 2017.

[9] C. Jia, Y. Yang, X. Zhang, S. Wang, S. Wang, and S. Ma, "Light field image compression with sub-apertures reordering and adaptive reconstruction," in *the Pacific-Rim Conference on Multimedia (PCM)*, 2017.

[10] C. Jia, Y. Yang, X. Zhang, S. Wang, X. Zhang, S. Wang, and S. Ma, "Optimized inter-view prediction based light field image compression with adaptive reconstruction," in *IEEE Int. Conf. Image Processing (ICIP)*, 2017.

[11] J. Chen, J. Hou, and L.-P. Chau, "Light field compression with disparity guided sparse coding based on structural key views," *IEEE Trans. Image Process.*, 2017.

[12] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, "Light field compression with homography-based low rank approximation," *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[13] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[14] G. Miller, S. Rubin, and D. Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," in *Rendering Techniques*. Springer, 1998, pp. 281–292.

[15] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3d photography," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 287–296.

[16] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light field mapping: efficient representation and hardware rendering of surface light fields," *ACM Trans. Graphics (TOG)*, vol. 21, no. 3, pp. 447–456, 2002.

[17] R. L. de Queiroz and P. A. Chou, "Compression of 3d point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3947–3956, 2016.

[18] R. Schnabel and R. Klein, "Octree-based point-cloud compression." *Spbg*, vol. 6, pp. 111–120, 2006.

[19] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.