Disparity Compensation Framework for Light-Field Coding Using Weighted Binary Patterns

Kohei Isechi†, Keita Takahashi (member)†, Toshiaki Fujii (member)†

Abstract We propose a disparity compensation framework for efficient light-field coding. We have proposed a novel light-field coding scheme of approximating a light field with the sum of weighted binary patterns. This coding scheme achieves comparable performance to the modern video coding standards H.265/HEVC and enables a dramatically simple decoding process, but its computational complexity for encoding is quite high. We have also proposed a progressive coding scheme that progressively encodes the target light field with a small number of weighted binary patterns in a step-by-step manner. The progressive scheme remarkably mitigates the computational complexity, but its rate-distortion performance is degraded. To address the above problems, we design a disparity compensation framework, which can be applied to the existing progressive coding, to improve the rate-distortion performance while keeping feasible computational complexity. Experimental results demonstrate that the proposed method improves the rate-distortion performance of the progressive coding without a computational complexity explosion.

Key words: Light-field coding, Progressive coding, Disparity compensation

1. Introduction

A light field, which has rich visual information in 3D space, is a fundamental data format for 3D image processing such as depth estimation1)−3), free-viewpoint image4)−6), and 3D display7)−9). A light field can be interpreted as a set of dense multi-view images, which can be captured by a light-field camera such as Lytro Illum10,11). A dense light field is composed of dozens to hundreds of images with very small viewpoint intervals in practical use.

Efficient light-field coding is one of the important research issues for light-field processing because the data amount of a dense light field is huge because of a large number of images. Most research uses modern video coding standards such as H.265/HEVC12) with small modifications for light fields13)−18); namely, video coding techniques such as intra/inter-frame prediction, discrete cosine transform (DCT), arithmetic coding, and so on, are applied to a light field to eliminate its redundancy. In particular, motion compensation techniques can be applied to remove redundancy among the images at different viewpoints in a dense light field.

In contrast, we believe that the standard video coding techniques are not necessarily the most suitable for a dense light field; then, we have proposed a novel light-field coding scheme using weighted binary patterns19,20). This coding scheme approximates a light field by using a linear combination of only several binary patterns as shown in Fig. 1. In this paper, this scheme is named “baseline” as a counterpart of the progressive scheme mentioned later. The binary patterns and weights are obtained by solving an optimization problem, of which an objective function is the difference between a target light field and a light field that is reconstructed as the sum of weighted binary patterns. The coding scheme achieves comparable rate-distortion performance to that of H.265/HEVC applied to light fields, and its decoding process, i.e., calculating the linear combination of binary patterns, is dramatically simpler than that of H.265/HEVC. However, the encoding process, which involves solving the optimization problem, takes much longer than the video coding standards. As the number of binary patterns increases, encoding time exponentially increases; then, the encoding time soon becomes infeasible. The encoding process can be made computationally lighter by dividing binary patterns and weights into several groups on the basis of the divide-and-conquer strategy as shown in Fig. 2. This framework progressively approximates a light field with a small number of weighted binary patterns at each group and can greatly reduce the computational complexity for obtaining optimal binary patterns. However, the approximation accuracy is slightly degraded.
in exchange for reduction of computational complexity. To summarize, the coding schemes presented in our previous work\textsuperscript{19,20} have a problem of the trade-off between computational complexity and rate-distortion performance.

One of the methods for dealing with the problem is the use of disparity compensation, which is to shift the pixels in the images according to a specified disparity value and the viewpoint positions. In the existing coding schemes\textsuperscript{19,20}, the binary patterns represent only the common components among the images at different viewpoints, while the weights represent viewpoint-dependent components of the light field. If binary patterns can also represent viewpoint-dependent components, the approximation accuracy of the coding scheme would be improved. The disparity compensation can provide the binary patterns with the capability to represent viewpoint-dependent components. We have investigated the effectiveness of applying disparity compensation to the coding scheme\textsuperscript{19,20} and presented the preliminary results\textsuperscript{21}. Experimental results show that the disparity compensation improves the approximation accuracy, but the disparity values were empirically determined beforehand. The appropriate disparity values depend on a captured scene, so a method of finding the appropriate disparity values should be provided in some way.

In this paper, we propose a method of applying disparity compensation to the progressive light-field coding with weighted binary patterns. We aim at improving rate-distortion performance of the existing progressive light-field coding by adaptively applying disparity compensation while avoiding infeasible computational complexity. When disparity compensation is applied to each of the binary patterns in a straightforward manner in the optimization problem, it is quite difficult to solve the problem because it includes three sets of unknown values: binary patterns, its corresponding weights, and disparity values applied to the binary patterns. By combining the progressive scheme with disparity compensation, we can solve the optimization problem at each group with brute-force search on a set of candidate disparity values because the number of binary patterns in each group is small. The proposed method finds the best disparity value at each group so that the encoding result adaptively uses good disparity values depending on target light fields. Furthermore, disparity compensation is not considered in the baseline scheme\textsuperscript{19,20}; thus, the proposed method might outperform not only the existing progressive framework but also the baseline one.

### 2. Light-field coding with weighted binary patterns

This section summarizes an existing light-field coding framework using weighted binary patterns\textsuperscript{19,20}. We assume that a light field, which is equivalent to multi-view images, to be compressed is given as $L(s, t, x, y)$. The set of symbols $(s, t), (s = 1, \ldots, S, t = 1, \ldots, T)$ and $(x, y), (x = 1, \ldots, X, y = 1, \ldots, Y)$ indicate viewpoint coordinates in a 2D grid and pixel coordinates in a viewpoint image, respectively. The baseline method approximates a light field $L(s, t, x, y)$ by using the sum of $N$ weighted binary patterns as follows:

$$L(s, t, x, y) \simeq \sum_{n=1}^{N} B_n(x, y)r_n(s, t), \quad (1)$$

where $B_n(x, y) \in \{0, 1\}, (n = 1, 2, \cdots, N)$ and $r_n(s, t) \in \mathbb{R}$ indicate binary patterns and corresponding weights, respectively. All the images at different viewpoints are represented with the same binary patterns, but different weights are used depending on the view-
points; therefore, the binary patterns and the weights have the common and viewpoint-dependent components of the multi-view images, respectively.

To obtain binary patterns and weights that can accurately approximate the light field, we solve an optimization problem defined as:

$$\arg \min_{B_n(x,y), r_n(s,t)} \sum_{i=1}^{N} ||B_n(x,y) - \sum_{n=1}^{N} B_n(x,y) r_n(s,t)||^2.$$  \hfill (2)

The optimal solution to Eq. (2) cannot be easily obtained because Eq. (2) includes two sets of unknowns, $B_n(x,y)$ and $r_n(s,t)$. We use an alternative optimization to obtain the solution, where the binary patterns $B_n(x,y)$ are initialized at first, and we repeat the two following steps until convergence. (i) The binary patterns $B_n(x,y)$ are fixed, and the weights $r_n(s,t)$ are optimized. This optimization can be regarded as a standard least squares minimization problem that can be easily solved by well-known methods. (ii) The weights $r_n(s,t)$ are fixed, and the binary patterns $B_n(x,y)$ are optimized. The solution to this optimization can be obtained individually for each pixel $(x,y)$ because this problem is pixel-independent. This optimization is regarded as a binary combinational optimization known as a NP-hard problem. We obtain its solution by using simple brute-force search. After convergence, a light field $L(s,t,x,y)$ can be reconstructed using Eq. (1) with the obtained solution to Eq. (2).

To accelerate the encoding process, we make the coding scheme progressive on the basis of the divide-and-conquer strategy\(^{(19)}\)\(^{(20)}\) as shown in Fig. 2. Assuming that an original light field $L(s,t,x,y)$ is approximated by using totally $N$ binary patterns and weights, the $N$ binary patterns are divided into $M$ groups (layers). Each layer has $N'$ binary patterns so that $N'M = N$ is satisfied. At the first layer, the target light field $L_1(s,t,x,y)$ is set as an original light field $L(s,t,x,y)$ and is approximated by using $N'$ binary patterns and weights based on Eq. (2). The light field obtained by the approximation is denoted as $L'_1(s,t,x,y)$. At the next layer, the target light field $L_2(s,t,x,y)$ is defined as the difference between $L(s,t,x,y)$ and $L'_1(s,t,x,y)$. The target light field $L_2(s,t,x,y)$, i.e., the residual light field, is also approximated with $N'$ binary patterns and weights in the same manner as the first layer. At the following layers, we progressively repeat the process where $N'$ binary patterns and weights are calculated to approximate the target light field $L_m(s,t,x,y)$ defined as:

$$L_m(s,t,x,y) = L(s,t,x,y) - \sum_{i=1}^{m-1} L'_i(s,t,x,y),$$  \hfill (3)

$$L'_i(s,t,x,y) = \sum_{n=1}^{N'} B_n^{(i)}(s,t)r_n^{(i)}(s,t),$$  \hfill (4)

where $B_n^{(i)}(s,t)$, $r_n^{(i)}(s,t)$, and $L'_i(s,t,x,y)$ denote the $n$-th binary pattern, the corresponding weight, and the approximated light field at the $i$-th layer, respectively. Finally, an original light field $L(s,t,x,y)$ is approximated as follows:

$$L(s,t,x,y) \simeq \sum_{m=1}^{M} L'_m(s,t,x,y).$$  \hfill (5)

This progressive extension achieves remarkable reduction of the computational complexity by optimizing a small number of binary patterns and weights per layer on the basis of the divide-and-conquer strategy. Although the strategy generally cannot bring about the global-optimum solution, a feasible solution can be obtained with less computational complexity since the search space of the problem is reduced. In the progressive coding, dividing the binary patterns and weights into $M$ groups is equal to the reduction of search space. The main bottleneck of the baseline scheme is optimizing binary patterns; thus, reducing the number of binary patterns in the optimization can significantly accelerate the encoding process. The computational complexity for the baseline scheme using $N$ binary patterns is $O(2^N)$, but the computational complexity for the progressive scheme is reduced to $O(M \cdot 2^{N'})$ by dividing $N$ binary patterns into $M$ layers. For instance, the computational cost of the progressive coding is reduced to 1/128 compared with that of the baseline one in the case with $N = 12$, $M = 4$, and $N' = 3$. As described above, the progressive extension finds a feasible solution with less computational complexity, but the approximation accuracy cannot help being degraded since the divide-and-conquer strategy cannot find the global-optimum solution. Consequently, there is a trade-off between computational complexity and rate-distortion performance.

When a target light field is represented by using $N$ binary patterns and weights, the compression ratio of the scheme is calculated as follows:

$$\text{compression ratio} = \frac{N(XY + STb_r)}{STXY b_L},$$  \hfill (6)

where $b_L$ and $b_r$ indicate bit depths used for light-field pixels $L(s,t,x,y)$ and weights $r_n(s,t)$, respectively. To further reduce the data amount, a lossless compression
algorithm, e.g., gzip, can be applied to the binary patterns and weights.

3. Disparity compensation for progressive light-field coding

This section expounds the proposed method, which introduces a disparity compensation framework to the progressive light-field coding scheme described in Section 2 to improve the rate-distortion performance while avoiding infeasible computational complexity. In the conventional scheme, all the images at different viewpoints are approximated using the same binary patterns with different weights. This means that the binary patterns represent the common components among multi-view images and that the weights represent viewpoint-dependent components. Meanwhile, if disparity compensation is applied to the binary patterns, they can represent not only the common components but also the viewpoint-dependent components. Thus, applying appropriate disparity compensation might improve the representation capability of the binary patterns. The amount of disparities included in a light field depend on a captured scene; therefore, the appropriate disparity values should be adaptively searched for according to the scene.

With disparity compensation applied, the approximation of a light field can be formulated as follows:

$$L(s, t, x, y) \simeq \sum_{n=1}^{N} B_n(x - sd_n, y - td_n) r_n(s, t),$$  \hspace{1cm} (7)

where $d_n$ is a disparity value with which the $n$-th binary pattern is compensated. All pixels of $B_n(x, y)$ are shifted according to the viewpoint position $(s, t)$ so that each viewpoint image is approximated by using slightly shifted binary patterns depending on the viewpoint position. According to Eq. (7), the optimization problem to find the binary patterns, weights, and disparity value is defined as follows:

$$\arg \min_{B_n(x, y), r_n(s, t)} \sum_{s, t, x, y} \| L(s, t, x, y) - L'(s, t, x, y) \|^2,$$  \hspace{1cm} (8)

$$L'(s, t, x, y) = \sum_{n=1}^{N} B_n(x - sd_n, y - td_n) r_n(s, t).$$  \hspace{1cm} (9)

Although it would be preferable if we can obtain the global-optimum solution for Eq. (8), solving this optimization is quite difficult because it includes three sets of unknowns. We reformulate Eq. (7) with a restriction on disparity values $d_n$ and then introduce a disparity compensation framework into the progressive light-field coding scheme presented in Section 2.

We propose a progressive light field coding method that is formulated as follows:

$$L(s, t, x, y) \simeq \sum_{m=1}^{M} L'_m(s, t, x, y)$$  \hspace{1cm} (10)

$$L'_m(s, t, x, y) = \sum_{n=1}^{N} B_n^{(m)}(x - sd_n, y - td_n) r_n^{(m)}(s, t),$$  \hspace{1cm} (11)

where $d_m$ is a disparity value used for the disparity compensation at the $m$-th layer. As a restriction, we use only one disparity value for each group of binary patterns. At each layer, binary patterns, weights, and a disparity value are obtained by solving the following optimization:

$$\arg \min_{B_n^{(m)}(x, y), r_n^{(m)}(s, t)} \sum_{s, t, x, y} \| L_m(s, t, x, y) - L'_m(s, t, x, y) \|^2,$$  \hspace{1cm} (12)

$$L_m(s, t, x, y) = L(s, t, x, y) - \sum_{i=1}^{m-1} L'_i(s, t, x, y).$$  \hspace{1cm} (13)

Equation (12) still includes three sets of unknowns like Eq. (8). The range of disparities included in a dense light field is basically narrow because of its very small viewpoint interval; thus, we simplify the problem by manually defining a set of candidate disparities $\mathcal{D}$.

The specific algorithm is shown in Algorithm 1, where $B_n^{(m)}(x, y)$, $r_n^{(m)}(s, t)$, and $d_n^{(m)}$ denote a solution for Eq. (12) at the $m$-th layer. We calculate the binary patterns and weights for each candidate disparity value $d \in \mathcal{D}$. From the set of the obtained binary patterns and weights, we use the one that achieves the best approximation accuracy. Consequently, the proposed method adaptively searches for the appropriate disparity value at each layer depending on a captured scene by simple brute-force search.

To practically solve Eq. (12), we reformulate it as follows:

$$\arg \min_{B_n^{(m)}(x, y), r_n^{(m)}(s, t)} \sum_{s, t, x, y} \| L_m(s, t, x' + sd_m, y' + td_m) - L'_m(s, t, x' + sd_m, y' + td_m) \|^2,$$  \hspace{1cm} (14)

$$L'_m(s, t, x' + sd_m, y' + td_m) = \sum_{n=1}^{N} B_n^{(m)}(x', y') r_n^{(m)}(s, t),$$  \hspace{1cm} (15)

where $x' = x - sd_m$, and $y' = y - td_m$. The right
side of Eq. (15) takes the same form as the second term in Eq. (2). We first apply disparity compensation to the target light field \( L_m(s, t, x, y) \) to obtain \( L_m(s, t, x' + sd, y' + td) \); after that, the solution for (14) is obtained in the same manner as solving Eq. (2). The desired light field \( L'(s, t, x, y) \) can be obtained by applying inverse disparity compensation to the light field \( L'(s, t, x', y' + td_m) \).

By manually defining a set of candidate disparity values, the proposed method finds the best disparity values while keeping feasible computational complexity. The difference between the proposed scheme and the conventional progressive method with respect to computational complexity is the brute-force search for disparity values. Assuming that the number of elements of \( \mathcal{D} \), i.e., the number of candidate disparity values, is denoted as \( D \), the computational complexity for optimizing binary patterns in the proposed scheme is \( O(\text{DM} \cdot 2^N) \) because the proposed method obtains the binary patterns and weights with all candidate disparity values at each layer. Although the proposed method takes much more time for encoding than the conventional progressive coding, it still can find the solution with feasible computational complexity. As we mentioned in Section 2, the main bottleneck is optimizing binary patterns; thus, if \( N \) is kept small, the computational complexity of the proposed method never increases explosively like that of the conventional baseline scheme as the total number of binary images \( N \) increases. In the case with \( N = 24 \), \( M = 8 \), \( N = 3 \), and \( D = 10 \), the proposed scheme takes 10 times longer than the conventional progressive scheme, but the computational cost of the proposed scheme \( 10 \times 8 \times 2^3 = 640 \) is still feasible compared with that of the baseline scheme.

When a target light field is represented by using \( N \) binary patterns and the corresponding weights, which are divided in \( M \) groups, the compression ratio of the proposed scheme is calculated as follows:

\[
\text{compression ratio} = \frac{N(XY + STb_d) + Mb_d}{STXYb_L},
\]

where \( b_d \) denotes a bit depth used to describe the used disparity values. The increase of total bits compared with Eq. (6) is less of an issue because one byte \( (b_d = 8) \) is enough to describe a disparity value when we set \( D \) to 10–20.

### 4. Experiment

We implemented our method using the software made available from our website. Six light-field datasets shown in Fig. 4, each of which consists of \( 17 \times 17 \) grayscale multi-view images, were used in the experiments. The number of iterations for alternative optimization to calculate binary patterns and weights was fixed at 20. The set of candidate disparity values \( \mathcal{D} \) in the proposed method was given as \( \mathcal{D} = \{0.0, \pm 0.2, \pm 0.5, \pm 0.8, \pm 1.0, \pm 1.5, \pm 2.0\} \); namely, the number of candidate disparity values is \( D = 13 \). All binary patterns \( B_n(x, y) \) were initialized by the result of binary thresholding to the most top-left image of the input light field. The number of binary patterns in each layer was set as \( N = 3 \). The approximation accuracy was evaluated using peak signal-to-noise ratio (PSNR), which is calculated from the mean square errors over all the viewpoints and pixels.

We first investigated the effectiveness of the disparity compensation framework in the progressive light-field coding scheme on the basis of rate-distortion performance. We compared the proposed method, the conventional progressive coding, and the baseline coding. The number of binary patterns \( N \) for the proposed method and the conventional progressive coding was varied from 3 to 24, but \( N \) for the baseline coding was limited from 3 to 12 because of the high computational complexity. The bitrate was calculated from raw binary patterns, weights, and used disparity values without gzip compression. Figure 5 shows rate-distortion curves for six datasets. The proposed method is called “Progressive + disp. comp.” in the results. Compared to the conventional progressive coding, the proposed method remarkably improves rate-distortion performance.

---

**Algorithm 1 Progressive light-field coding using disparity-compensated and weighted binary patterns**

Input: \( L(s, t, x, y) \), \( \mathcal{D} \)

Output: \( B_{(m)}^{(n)}(x, y), B_{(m)}^{(n)}(s, t), d_{(n)}^{(m)} \) \( (n = 1, \ldots, N, m = 1, \ldots, M) \)

Initialize: \( L(s, t, x, y) \leftarrow L(s, t, x, y) \)

for \( i = 1 \) to \( M \) do

for each disparity \( d \in \mathcal{D} \) do

Obtain \( B_{(i)}^{(n)}(x, y), B_{(i)}^{(n)}(s, t) \) using Eq. (12) with fixed \( d_{(n)} = d \)

\( p \leftarrow \text{PSNR of } L(s, t, x, y) \) from Eq. (10) with \( M = i \)

if \( \text{BEST-PSNR} < p \) then

\( B_{(i)}^{(n)}(x, y) \leftarrow B_{(i)}^{(n)}(x, y) \)

\( v_{(i)}^{(n)}(s, t) \leftarrow v_{(i)}^{(n)}(s, t) \)

\( d_{(n)}^{(m)} \leftarrow d \)

\( \text{BEST-PSNR} \leftarrow p \)

end if

end for

Carry over the residual using Eq. (13) with \( m = i + 1 \)

\( i \leftarrow i + 1 \)

end for

---
performances for truck and bulldozer while showing almost the same performances for the other datasets. The rate-distortion performances of the proposed method for truck and bulldozer even outperform those of the baseline method. Table 1 indicates selected disparity values in the proposed method with $N = 24$ for each layer of six datasets. The proposed method finds and uses non-zero disparity values in most of the layers for truck and bulldozer; thus, the disparity compensation outstandingly makes a difference for the two datasets. Meanwhile, for the other datasets, the selected disparities were zero for almost all the layers, which explains the reason why our method and the progressive method performed similarly in Fig. 5; these datasets have only small disparities, and thus, the progressive method without disparity compensation was sufficiently effective for them. Figure 4 presents visual comparisons between the proposed method and the conventional progressive coding. It seems that the proposed method achieves better approximation accuracy by alleviating blurs on the object’s parts having large disparities. Tables 2 and 3 show the comparison of encoding time of three methods for truck and knight datasets, respectively. The encoding time was measured on a desktop PC running Windows 10 Pro equipped with an Intel Core (TM) i7-6700 3.4-GHz CPU and 16.0-GB main memory. The proposed method takes much more time for encoding than the conventional progressive method as described in Section 3. However, the encoding time of the proposed method linearly increases as the number of binary patterns increases, while the encoding time of the baseline method explosively increases. The experimental results prove that the proposed method improves rate-distortion performance while avoiding a computational complexity explosion.

We next compared the proposed method with the modern video coding standard H.265/HEVC. As general implementations of H.265/HEVC, we used the HEVC Test Model (TM) i7-6700 3.4-GHz CPU and 16.0-GB main memory. The proposed method takes much more time for encoding than the conventional progressive method as described in Section 3. However, the encoding time of the proposed method linearly increases as the number of binary patterns increases, while the encoding time of the baseline method explosively increases. The experimental results prove that the proposed method improves rate-distortion performance while avoiding a computational complexity explosion.

We next compared the proposed method with the modern video coding standard H.265/HEVC. As general implementations of H.265/HEVC, we used the HEVC Test Model (TM) i7-6700 3.4-GHz CPU and 16.0-GB main memory. The proposed method takes much more time for encoding than the conventional progressive method as described in Section 3. However, the encoding time of the proposed method linearly increases as the number of binary patterns increases, while the encoding time of the baseline method explosively increases. The experimental results prove that the proposed method improves rate-distortion performance while avoiding a computational complexity explosion.

We next compared the proposed method with the modern video coding standard H.265/HEVC. As general implementations of H.265/HEVC, we used the HEVC Test Model (TM) i7-6700 3.4-GHz CPU and 16.0-GB main memory. The proposed method takes much more time for encoding than the conventional progressive method as described in Section 3. However, the encoding time of the proposed method linearly increases as the number of binary patterns increases, while the encoding time of the baseline method explosively increases. The experimental results prove that the proposed method improves rate-distortion performance while avoiding a computational complexity explosion.

We next compared the proposed method with the modern video coding standard H.265/HEVC. As general implementations of H.265/HEVC, we used the HEVC Test Model (TM) i7-6700 3.4-GHz CPU and 16.0-GB main memory. The proposed method takes much more time for encoding than the conventional progressive method as described in Section 3. However, the encoding time of the proposed method linearly increases as the number of binary patterns increases, while the encoding time of the baseline method explosively increases. The experimental results prove that the proposed method improves rate-distortion performance while avoiding a computational complexity explosion.
Fig. 5 R-D curves of the proposed method (Progressive + disp. comp.), the conventional progressive method (Progressive), and the conventional baseline method (Baseline).

Fig. 6 Comparison with modern video coding standard H.265/HEVC

| Table 2 | Encoding time for Truck [s] |
|-------------------|-----------------|
| # of binary patterns | 3   | 6   | 9   | 12  |
| Baseline           | 16.9 | 106  | 876 | 7521 |
| Progressive        | 16.8 | 32.0 | 49.0 | 64.1 |
| Progressive + disp. comp. | 263  | 514  | 774 | 1026 |

Table 3 Encoding time for Knight [s]

| # of binary patterns | 3   | 6   | 9   | 12  |
|-------------------|-----------------|
| Baseline           | 13.2 | 94.6 | 782 | 6768 |
| Progressive        | 13.5 | 26.0 | 39.4 | 52.9 |
| Progressive + disp. comp. | 212  | 422  | 643 | 852 |

46
and FFmpeg ver. 4.1. To apply these video codecs to light-field datasets, we aligned images in the dataset in the row-major order and regarded them as a video sequence. The bitrate of the proposed method was calculated from the binary patterns, weights, and used disparity values, which are compressed by gzip ver. 1.9. As a reference, the baseline method was also compared with the above methods. Figure 6 shows rate-distortion performances of three methods for six datasets. The proposed method shows better performance than that of the baseline method in truck dataset, and the performance can be more comparable to that of the HEVC Test Model. However, the proposed method still shows slightly inferior performance to the HEVC Test Model in the other datasets. We believe that the performance of the proposed method is promising because it achieves superior or comparable performance to that of FFmpeg and the HEVC Test Model, which has been optimized using enormous labor and time. The excellent performance of HEVC comes from the combination of many sophisticated coding techniques such as intra/inter prediction, transform coding, and arithmetic coding, where the optimal coding modes (e.g., prediction mode and block partition) are selected in accordance with the image content. Meanwhile, our method is merely constructed on a very simple framework using weighted binary patterns with disparity compensation.

5. Conclusion

In this paper, we proposed a method of introducing a disparity compensation framework into the progressive light-field coding with weighted binary patterns\(^{[10][19]}\). The proposed method adaptively finds the best disparity value at each layer by brute-force search on a manually-defined set of candidate disparity values. Experimental results show that, compared with the conventional progressive light-field coding, the proposed method improves the approximation accuracy in several datasets thanks to the disparity compensation. The proposed method outperforms not only the existing progressive method but also the original baseline method for some datasets. Although the proposed method takes much more time for encoding than the conventional progressive coding, the computational complexity never increases explosively like that of the conventional baseline coding.

For future work, we will also apply the disparity compensation framework to the baseline coding method and will extend the proposed method to larger images such as light fields captured with Lytro Illum cameras.

References

1) S. Wanner and B. Goldluecke, “Variational light field analysis for disparity estimation and super-resolution,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 3, pp. 606–619, March 2014.
2) T. Suzuki, K. Takahashi, and T. Fujii, “Sheared EPI analysis for disparity estimation from light fields,” IEICE Transactions on Information and Systems, vol. E100-D, no. 9, pp. 1984–1993, 2017.
3) G. Houben, S. Fujita, K. Takahashi, and T. Fujii, “Fast and robust disparity estimation for noisy light fields,” in 2018 25th IEEE International Conference on Image Processing (ICIP), Oct 2018, pp. 2010–2014.
4) M. Levoy and P. Hanrahan, “Light field rendering,” in Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ser. SIGGRAPH ’96. New York, NY, USA: ACM, 1996, pp. 31–42.
5) L. Shi, H. Hassannejad, A. Davis, D. Katabi, and F. Durand, “Light field reconstruction using sparsity in the continuous Fourier domain,” ACM Trans. Graph., vol. 34, no. 1, pp. 12:1–12:13, Dec. 2014.
6) J. Flynn, I. Neulander, J. Philbin, and N. Snavely, “Deepstereo: Learning to predict new views from the world’s imagery,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5515–5524, June 2016.
7) G. Wetzstein, D. Lannan, M. Hirsch, and R. Raskar, “Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting,” ACM Trans. Graph., vol. 31, no. 4, pp. 80:1–80:11, Jul. 2012.
8) T. Saito, Y. Kobayashi, K. Takahashi, and T. Fujii, “Displaying real-world light fields with stacked multiplicative layers: Requirement and data conversion for input multiview images,” Journal of Display Technology, vol. 12, no. 11, pp. 1290–1300, Nov 2016.
9) S. Lee, C. Jang, S. Moon, J. Cho, and B. Lee, “Additive light field displays: Realization of augmented reality with holographic optical elements,” ACM Trans. Graph., vol. 35, no. 4, pp. 60:1–60:13, Jul. 2016.
10) E. H. Adelson and J. Y. A. Wang, “Single lens stereo with a plenoptic camera,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 99–106, Feb 1992.
11) R. Ng, M. Levoy, M. Brédif, D. Guval, 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.
12) “PITU Recommendation H.265 and ISO/IEC 23008-2: High efficiency video coding,” Feb, 2018.
13) G. Tech, Y. Chen, K. Müller, J. R. Ohm, A. Vetro, and Y. K. Wang, “Overview of the multiview and 3D extensions of high efficiency video coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 1, pp. 35–49, Jan 2016.
14) Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, “Efficient intra prediction scheme for light field image compression,” in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2014, pp. 539–543.
15) A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao, “Data formats for high efficiency coding of Lytro-Illum light fields,” in 2015 International Conference on Image Processing Theory, Tools and Applications (IPTA), Nov 2015, pp. 494–497.
16) D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, “Pseudo-sequence-based light field image compression,” in 2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW), July 2016, pp. 1–4.
17) F. Hawary, C. Guillemot, D. Thoreau, and G. Boisson, “Scalable light field compression scheme using sparse reconstruction and restoration,” 2017 IEEE International Conference on Image Processing (ICIP), pp. 3250–3254, 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01597536.
18) X. Jiang, M. L. Pendu, R. A. Farrugia, and C. Guillemot, “Light field compression with homography-based low-rank approximation,” IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 7, pp. 1332–1345, Oct 2017.
19) K. Komatsu, K. Takahashi, and T. Fujii, “Scalable light field coding using weighted binary images,” 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 903–907.
20) K. Komatsu, K. Isechi, K. Takahashi, and T. Fujii, “Light field coding using weighted binary images,” *IEICE Transactions on Information and Systems*, vol. E102-D, no. 11, pp. 2110–2119, 2019.

21) K. Komatsu, K. Takahashi, and T. Fujii, “Multi-view image coding using disparity-compensated and weighted binary patterns,” in *Proc. of International Workshop on Advanced Image Technology (IWAIT) 2019*, vol. 11049, 2019.

22) “Light field compression project,” http://www.fujii.nuee.nagoya-u.ac.jp/~takahasi/Research/LFCompression/index.html, 2018. [Online]. Available: http://www.fujii.nuee.nagoya-u.ac.jp/~takahasi/Research/LFCompression/index.html

23) “The (new) stanford light field archive,” http://lightfield.stanford.edu/lfs.html, 2008.

24) “Fraunhofer heinrich hertz institute high efficiency video coding (hevc),” https://hevc.hhi.fraunhofer.de, 2018.

**Kohei Isechi** received his B.E. degree from the National Institution for Academic Degrees and Quality Enhancement of Higher Education, Japan, in 2015, and his M.E. degree in electrical engineering and computer science from Nagoya University, Japan, in 2017. He is working on 3D video compression. He is currently a Ph.D. student in information and communication engineering at Nagoya University.

**Keita Takahashi** received B.E., M.S., and Ph.D. degrees in information and communication engineering from the University of Tokyo, Tokyo, Japan, in 2001, 2003, and 2006. He was a project assistant professor at the University of Tokyo from 2006 to 2011 and was an assistant professor at the University of Electro-Communications from 2011 to 2013. He is currently an associate professor at the Graduate School of Engineering, Nagoya University, Japan. His research interests include computational photography, image-based rendering, and 3D displays.

**Toshiaki Fujii** received B.E., M.E., and Dr.E. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1990, 1992, and 1995. In 1995, he joined the Graduate School of Engineering, Nagoya University, where he is currently a professor. From 2008 to 2010, he was with the Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests include multi-dimensional signal processing, multi-camera systems, multi-view video coding and transmission, free-viewpoint television, and their applications.