

バイナリパターンの重み付け和による多視点画像の圧縮符号化

Compression of Multi-View Images Using Weighted Binary Patterns

小松滉治† 高橋桂太† 藤井俊彰†
Koji KOMATSU Keita TAKAHASHI Toshiaki FUJII

1 はじめに

異なる視点から同一の被写体を撮影した多視点画像は、三次元形状の計測や立体映像の生成など、様々な用途で用いられている。従来の多視点画像は、複数のカメラを並べて撮影されることが多いため、視点間隔が広く、隣り合う画像間の視差が数十画素にも及ぶ場合もあった。一方、近年では Light field camera の研究の発展 [1, 2, 3, 4, 5, 6] に伴い、視点の数は非常に多いものの、視点間隔が狭く、隣り合う画像間の視差が高々数画素になるような多視点画像が用いられるようになった [7, 8, 9, 10, 11]。本稿では、上記のような多視点画像を高密度多視点画像と呼ぶことにする。

本研究では、高密度多視点画像に適した圧縮手法を検討する。従来の画像圧縮の枠組みでは、画像間の予測と直交変換の組み合わせが長年の標準規格で用いられている [12]。多視点画像においては、画像間の対応領域を効率的に圧縮するため、視差補償予測やデプスマップによる予測が用いられてきた [13]。これらのアプローチは、画像間の視差が大きい場合に圧縮効率を高めるのに有効であるが、本研究が対象とする高密度多視点画像に対しては必ずしも最適であるとは限らない。また、デコード処理が複雑かつ計算コストが高いという欠点がある。

一方、提案手法では、少数の 2 値画像と重みの組み合わせによって高密度多視点画像を表現する。この表現法は、アクティブシャッター式メガネの映像表示のために考案された手法 [14] に着想を得たものである。これは、視差の小さな画像群が共通の 2 値画像によって表現可能であるという仮定に基づく。2 値画像は 1 画素あたり 1 bit で表現できるため大幅なデータ量の削減が見込める。実際、提案手法では、画像間の予測と直交変換を組み合わせた従来の符号化手法に匹敵する圧縮性能が得られる。さらに、提案手法では、単純な積和演算のみで多視点画像が復元できるため、デコード処理が簡便かつ高速であるという利点がある。

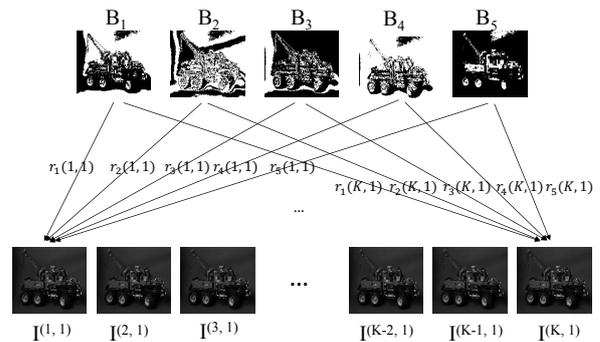


Fig. 1: Compression using weighted binary patterns

2 提案手法

本章では、 160×120 画素、 17×17 視点の多視点画像 Truck[15] を例に、提案手法を説明する。

2.1 2 値画像による多視点画像の表現

単純のため、グレースケール画像の場合を考える。図 1 に示すように、提案手法では、 M (横視点数 $K \times$ 縦視点数 L) 枚の多視点画像 $I^{(k,l)}(x,y)$ ($k = 1, 2, \dots, K, l = 1, 2, \dots, L$) を、 N 枚の 2 値画像 $B_n(x,y)$ ($n = 1, 2, \dots, N$) と $M \times N$ 個の重み $r_n(k,l)$ を組み合わせ、

$$I^{(k,l)}(x,y) \simeq \sum_{n=1}^N B_n(x,y) r_n(k,l) \quad (1)$$

$$B_n(x,y) \in \{0, 1\}, r_n(k,l) \in R$$

と表現する。この表現には 2 つの利点がある。まず、提案手法によって伝送されるのは、 N 枚の 2 値画像 $B_n(x,y)$ と $M \times N$ 個の重み $r_n(k,l)$ である。伝送に必要なビット数は、後述するように、 M 枚の多視点画像 $I^{(k,l)}(x,y)$ を直接伝送する場合と比べて大幅に削減される。また、 $B_n(x,y)$ と $r_n(k,l)$ との単純な積和演算により $I^{(k,l)}(x,y)$ を復元できるため、デコード処理が極めて高速である。

† 名古屋大学 大学院工学研究科
Nagoya University School of Engineering

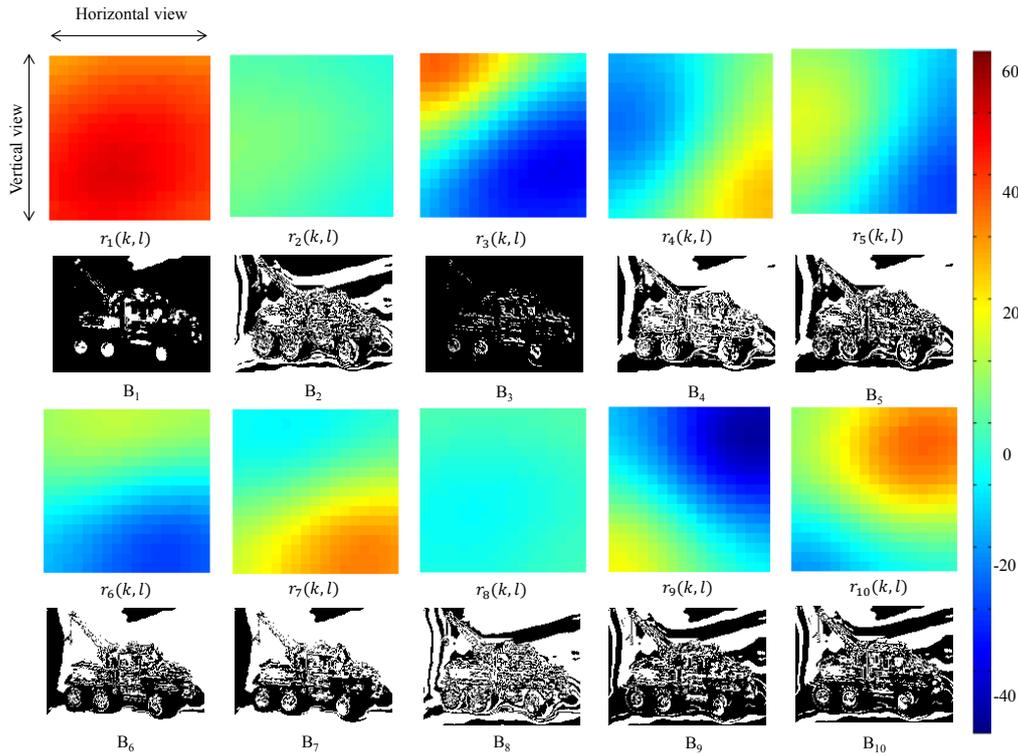


Fig. 2: Binary images and corresponding weights

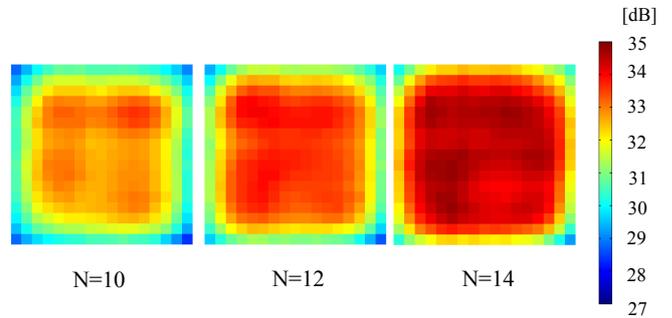
2.2 2値画像と重みの最適化

提案手法のエンコード処理は、多視点画像 $I^{(k,l)}(x,y)$ を入力として、(1) 式を満たすような2値画像 $B_n(x,y)$ と重み $r_n(k,l)$ を求めることである。そのためには、次式の最小二乗問題を解けばよい。

$$\arg \min_{B_n(x,y), r_n(k,l)} \sum_{x=1, y=1}^{w, h} \sum_{l=1}^L \sum_{k=1}^K |I^{(k,l)}(x,y) - \sum_{n=1}^N B_n(x,y) r_n(k,l)|^2 \quad (2)$$

ここで、多視点画像の画像サイズを $w \times h$ 画素とする。この式では2組の未知数 $B_n(x,y)$ と $r_n(k,l)$ があるため、適当な2値画像 $B_n(x,y)$ を初期値として与えた上で、一方を固定して他方を最適化する処理を交互に繰り返す。(i) まず、2値画像 $B_n(x,y)$ を固定し、重み $r_n(k,l)$ を最適化する。これは標準的な最小二乗問題となり、線形方程式の求解に帰着される。(ii) 次に、重み $r_n(k,l)$ を固定して2値画像 $B_n(x,y)$ を最適化する。これは、画素 (x,y) 毎に N 個の2値パターンの組み合わせを最適化する問題となり、 $O(2^N)$ の計算コストを要する。(i) と (ii) の処理を取束するまで交互に繰り返す。

上記の計算によって得られた2値画像 $B_n(x,y)$ と各視点に対する重み $r_n(k,l)$ を図2に示す。ここで、2値画像の枚数 N は10とした。2値画像の初期値は、1枚

Fig. 3: PSNR of each view image with different N

目の入力画像を平均画素値を閾値として2値化した画像とした。この初期値の設定を入力画像閾値法と呼ぶことにする。最適化の繰り返しは10回とした。同一の初期値からスタートしたにも関わらず、10枚の2値画像はそれぞれ異なるパターンとなった。また、各2値画像に対する重みを観察すると、視点に応じて滑らかに重み変化していることが分かる。2値画像の枚数 N を変化した場合の視点毎のPSNRを図3に示す。 N が大きくなるほど全体の品質が向上することが分かる。また、中央の視点ほど品質が高く周辺視点では品質が低い。これは、中央付近の視点になるほど他の視点との共通部分が多いためだと考えられる。

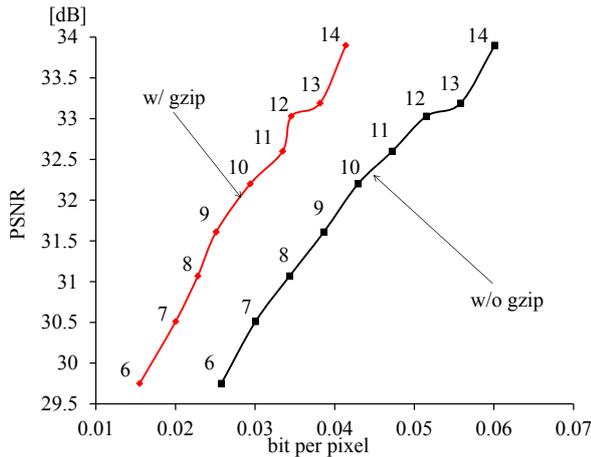


Fig. 4: R-D curves of proposed method w/ and w/o gzip

2.3 提案手法の圧縮率

入力となる多視点画像 $I^{(k,l)}(x,y)$ のデータ量は、画像の枚数を M 枚、画像のサイズを $w \times h$ 画素、1 画素あたりの階調 d bit とすると、 $Mwhd$ bit である。一方、提案手法により伝送されるデータ量は、 N 枚の 2 値画像に対して Nwh bit、 MN 個の重みに対して aMN bit となる。ここで a は重みを表現するビット深度である。提案手法の圧縮率は以下の式で与えられる。

$$\text{compression ratio} = \frac{Nwh + aMN}{Mwhd} \quad (3)$$

本研究では、表現する対象の多視点画像のビット深度が $d=8$ のため、重みについても $a=16$ 程度の精度で充分であった。そこで、重みを定数倍して short int 型で保存することとした。例として、 $M=289$, $w=160$, $h=120$ の多視点画像に対して $N=10$ として (3) 式を計算すると、圧縮後のデータ量は元データの約 0.54% となる。また、2 値画像と重みをバイナリファイルで保存し、そのファイルに対して gzip などの他の圧縮手法を適用すれば、さらにデータ量を小さくできる。例えば、Truck のデータの場合、0.32% となった。

2.4 最適化演算の高速化

提案手法では、デコード処理は単純な積和演算のため高速だが、エンコード処理は最適化演算となるため処理コストが高い。ここでは、最適化演算を以下の 2 つのステップで高速化する。(i) 元の問題を分割することにより、近似解を高速に求める。(ii) 得られた近似解を初期値として用いることで、目的とする解に達するまでの繰り返し回数を削減する。

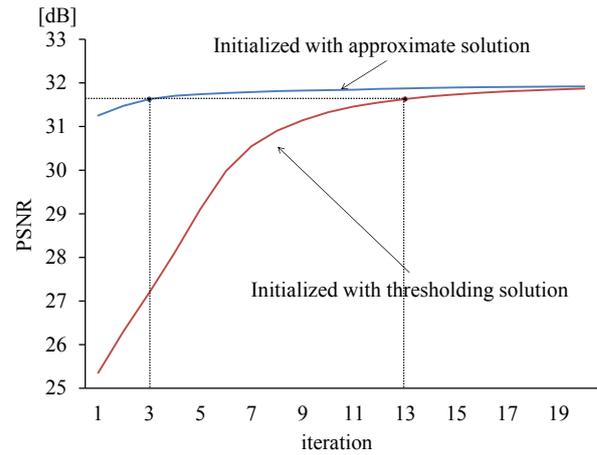


Fig. 5: Number of iterations and PSNR

Tab. 1: Computation time for optimization

| | 繰り返し [回] | 時間 [s] |
|---------------|----------|--------|
| (i) 近似解の導出 | | 103 |
| 平均画像復元 | 38 | 25 |
| 差分画像復元 | 10 | 78 |
| (ii) 初期値:近似解 | 3 | 320 |
| (i)+(ii) 合計時間 | | 423 |
| 初期値:入力画像閾値法 | 13 | 1376 |

まず、近似解の求め方を述べる。元の問題における 2 値画像の枚数を N とする。最適化において最も計算コストを要するのは 2 値パターンの最適化であり、 $O(2^N)$ である。一方、元の問題を N_1 枚の 2 値画像に対する最適化問題と N_2 枚の 2 値画像に対する最適化問題に分割することを考える。ただし、 $N = N_1 + N_2$ である。この時、2 値パターンの最適化の計算オーダーは $O(2^{N_1}) + O(2^{N_2})$ である。例として $N=10$, $N_1=5$, $N_2=5$ とした場合、元の問題の計算コストは 1024 であり、分割された問題の計算コストは 64 である。したがって、分割された問題は非常に小さなコストで解けることがわかる。

具体的な問題分割の例を述べる。多視点画像 $I^{(k,l)}(x,y)$ の平均画像を $I(x,y)$ 、各視点画像と平均画像の差分を $\tilde{I}^{(k,l)}(x,y) = I^{(k,l)}(x,y) - I(x,y)$ とする。平均画像 $I(x,y)$ を N_1 枚の 2 値画像で、 M 枚の差分画像 $\tilde{I}^{(k,l)}(x,y)$ を N_2 枚の 2 値画像で表す問題を考える。分割された問題の初期値は、入力画像閾値法で定めた。それぞれの問題から得られる解を組み合わせると元の問題の近似解となる。Truck を用いた実験では、元の問題の 1/13 程度の時間で元の問題を直接解いた場合と比較して -1 dB

程度の近似解を得ることができた。

次に、近似解を初期値として設定し、元の問題を解く。図 5 に初期値を近似解とした場合、および入力画像閾値法で定めた場合の解の収束の様子を示す。横軸が最適化の繰り返し回数、縦軸が復元画像の PSNR を表す。このグラフから初期値として近似解を用いた方が少ない繰り返し回数で収束することがわかる。近似解を用いた場合、例えば繰り返し回数を 3 回で止めても PSNR は 31.6 dB となり、直接解法（入力画像閾値法で初期値を定める場合）で 13 回繰り返した時と同等の品質が得られる。この時、近似解を求める処理を含めても、高速化手法の処理時間は直接解法の約 30% となり、高速化が達成されている。高速化手法および直接解法の処理時間および最適化演算の繰り返し回数を表 1 にまとめる。

2.5 カラー画像への拡張

(1) 式をカラー画像へ拡張する方法はいくつか考えられる。まず、RGB のチャンネル毎に独立に (1) 式を当てはめる方法がある。この方法では 3 チャンネル分の最適化計算が必要になる。また、色チャンネル間の相関を活用した圧縮表現にはならない。別の方法として色空間を YCbCr に変換し、変換後のチャンネルに独立に (1) 式を当てはめる方法も考えられる。この方法では色変換によってチャンネル間の相関は除去できるが、変換後の各チャンネルにそれぞれ何枚の 2 値画像を割り当てるかに最適化の余地があり、必ずしも扱いやすくない。そこで我々は、色チャンネルを RGB のまま扱うこととし、チャンネル間の相関を考慮するため、全てのチャンネルで 2 値画像 $B_n(x, y)$ を共有する以下のモデルを考える。

$$I^{(k,l,c)}(x,y) \simeq \sum_{n=1}^N B_n(x,y)r_n(k,l,c) \quad (4)$$

$$B_n(x,y) \in \{0,1\}, r_n(k,l,c) \in R$$

ここで、 $c \in \{0,1,2\}$ はカラーチャンネルの番号を表す。(4) 式は、左辺の画像の枚数および右辺の重みの数が 3 倍になることを除けば、(1) 式と全く等価な形式である。したがって、最適化手法も全く同一となる。ただし、圧縮率は、元が多視点画像が 3 チャンネル分、重みも 3 チャンネル分あることを考慮すると、以下のようになる。

$$\text{compression ratio} = \frac{Nwh + 3aMN}{3Mwhd} \quad (5)$$

図 6 に (1) 式のモデルで RGB チャンネルを独立に扱う場合と、(4) 式のモデルで RGB チャンネル間で 2 値画像

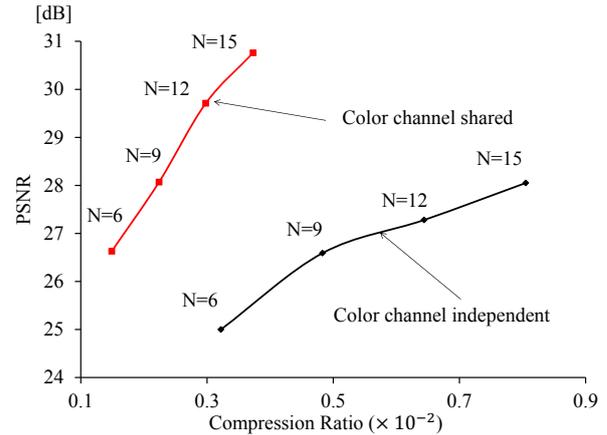


Fig. 6: R-D curves of color multi-view images

Tab. 2: experiment environment

| | |
|--------|----------------------------------|
| CPU | Intel(R) Core(TM) i7-4790 3.60GB |
| OS | Virtual Box Ubuntu 16.04 |
| メモリ | 2048MB |
| ffmpeg | ver. 2.8.11 |
| gzip | ver. 1.6 |

を共有して扱う場合のレート歪み特性を示す。前者では 2 値画像の枚数は各チャンネルで同一にした。横軸は (3) 式または (5) 式で求められる圧縮率、縦軸は画像全体の復元品質を示している。図中のプロットには用いた 2 値画像の合計枚数を併記している。グラフから、色チャンネル間で 2 値画像を共有することによって大幅に圧縮性能が上がる事がわかる。

3 性能評価

3.1 提案手法と従来の映像符号化手法の比較

提案手法と既存の映像符号化手法を比較するため、図 7 に示したデータセットを用いて実験を行った。データセットはいずれも縦横 17×17 視点の合計 289 枚、グレースケールの多視点画像を用いた。詳細な実験環境は表 2 に示す。

従来の映像符号化手法を実装したツールとして ffmpeg を用いた。多視点画像を左上視点から右下視点へ行優先順に並べ、ビデオデータとみなして符号化した。ffmpeg では H.264 と HEVC がサポートされており、画像間での予測を用いた効率的な符号化が実現できる。H.264 ではオプションによって GOP を変更できるが、例えば GOP を横方向視点数 17 の倍数に設定するなど

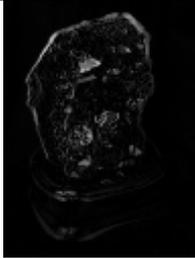
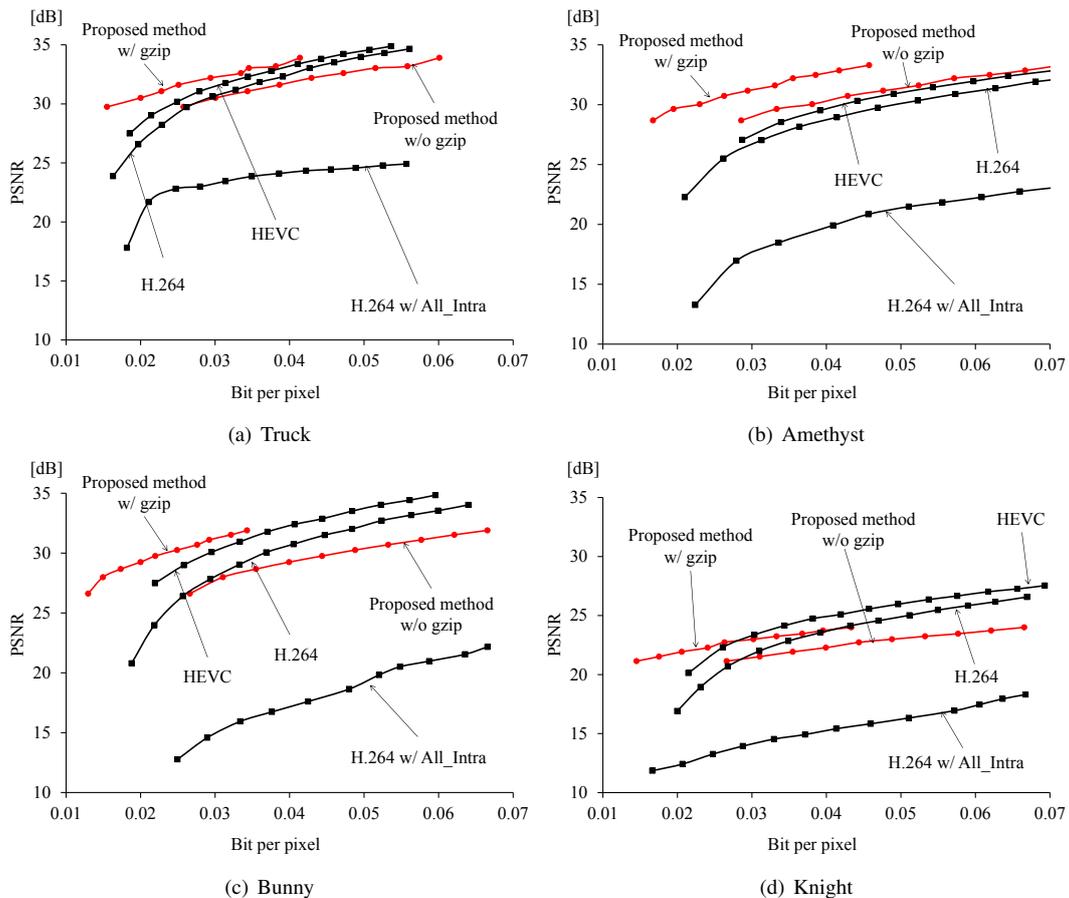
| Datasets | Truck | Amethyst | Bunny | Knight |
|---|---|---|--|---|
| # of views | 17×17 | | | |
| size | 160×120 | 96×128 | 128×128 | 128×128 |
| top left image |  |  |  |  |
| Difference between top left and top right image |  |  |  |  |

Fig. 7: Datasets


 Fig. 8: R-D curves of proposed method and conventional video codecs with 17×17 viewpoints datasets

工夫しても性能が上がりなかつたため、GOP はソフトウェアの自動設定に任せた。HEVC では GOP を設定するオプションが実装されていなかった。また、H.264 の場合はオプションを指定することで全てのフレームをイントラフレーム方式で圧縮することもできる。画像間の予測の有無による性能の違いを見るため、全てイントラフレームで圧縮する場合も比較に含めた。さらに、ffmpeg では、動画の閲覧を前提とした高速なデコードが実装されているため、提案手法とのデコード時間の比較も可能である。

まず、図 8 に提案手法と従来の映像符号化手法のレート歪み曲線を示す。提案手法では、2.4 節で述べた高速化手法は使わず、最適化の繰り返し演算回数を Truck では 20 回としその他のデータセットでは 10 回に固定した。また、2 値画像と重みのバイナリファイルをさらに gzip で圧縮した場合も評価に加えた。提案手法では用いる 2 値画像の枚数を変えることでビットレートを制御し、従来の符号化方式では ffmpeg にオプションとしてビットレートを与えた。図 8 を見ると、データセットによって傾向は異なるものの、提案手法と gzip を組み合わせると、HEVC や H.264 に匹敵する、あるいは上回るレート歪み性能が得られることがわかる。また、H.264 において全てイントラフレームとした場合にはレート歪み特性が極端に悪化することから、画像間の予測が圧縮に有効に効いていることがわかる。一方、提案手法では、従来手法において画像間予測を活用した場合と同等以上の性能を、共通の 2 値画像と画像ごとの重みの組み合わせで達成できている。

次に、図 9 に提案手法と従来の映像符号化手法のデコード時間を示す。提案手法では、gzip ファイルを解凍し、2 値画像と重みのバイナリファイルから多視点画像を復元した後、画像をディスク上に保存するまでの時間を計測した。従来の映像符号化手法では、動画ファイルを読み込んで展開し、多視点画像としてディスク上に保存するまでの時間を計測した。提案手法では $N = 14$ とし、従来の映像符号化手法では PSNR が提案手法と同等になるようにビットレートを設定した。図 9 に示す結果は、time コマンド使って 10 回計測した平均値である。この図より、提案手法は従来の映像符号化手法よりもデコードが高速であり、gzip の解凍を含めても HEVC の約 20 % の時間で済むことがわかる。また、提案手法のデコード時間のうち半分以上がディスク上へのファイルの書き込みの時間であるため、多視点画像の復元自体は非常に高速であることがわかる。これは、提案手法では、デコード処理が単純な積和演算のみであり、画像間予測や変換を必要とする従

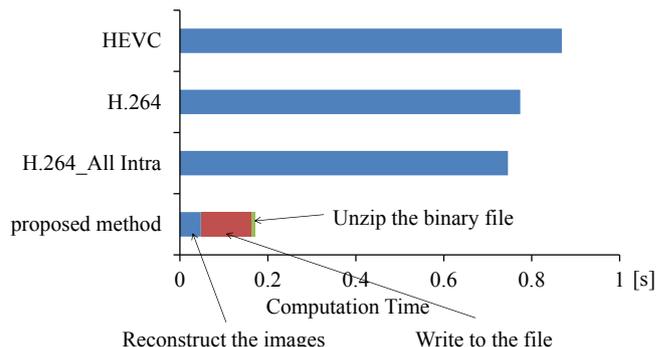


Fig. 9: Comparison of decoding time

来の映像符号化手法よりもはるかに計算コストが低いことに対応する。

3.2 提案手法における視点間隔の影響

図 8 の実験において、Knight データセットに対しては提案手法が従来の映像符号化手法に劣っていることがわかる。これは、図 7 の差分画像からわかるように、Knight では左右の視点の画像の違いが大きいためだと考えられる。したがって、提案手法は、画像間の違いが小さい多視点画像に対して有効だが、画像間の違いが大きくなるほど不利になると考えられる。

この仮説を検証するため、同一の被写体に対して視点間隔を変更し、画像間の違いをコントロールしつつ実験を行った。具体的には、Truck データセットにおいて図 10(a)–(d) に示すように視点間隔を変えた $5 \times 5 = 25$ 枚を選択し、それぞれを Dataset A–D とした。図 10(e)(f) に、提案手法と HEVC を用いた場合のそれぞれの Dataset に対するレート歪み曲線を示す。ここで、提案手法では、視点間隔の影響を明確にするため gzip を適用せず、2 値画像と重みのバイナリファイルの容量をそのままビットレートに換算した。

これらの図より、どちらの手法においても視点間隔が小さいほどレート歪み特性が良いことがわかる。これは、視点間隔が小さいほど画像間の違いが小さくなることに対応しており、自然な結果である。しかし、視点間隔が大きくなった時のレート歪み特性の劣化の度合いは提案手法の方が大きい。提案手法では、共通の 2 値画像によって多視点画像を表現するため、画像間の違いが大きくなると復元性能が顕著に低下すると考えられる。一方で、画像間予測を用いる HEVC では、視点間隔が大きくなり視差が大きくなったとしてもレート歪み特性をある程度維持できる。したがって、提案手法が有効となるのは、視点間隔が十分に小さい多視

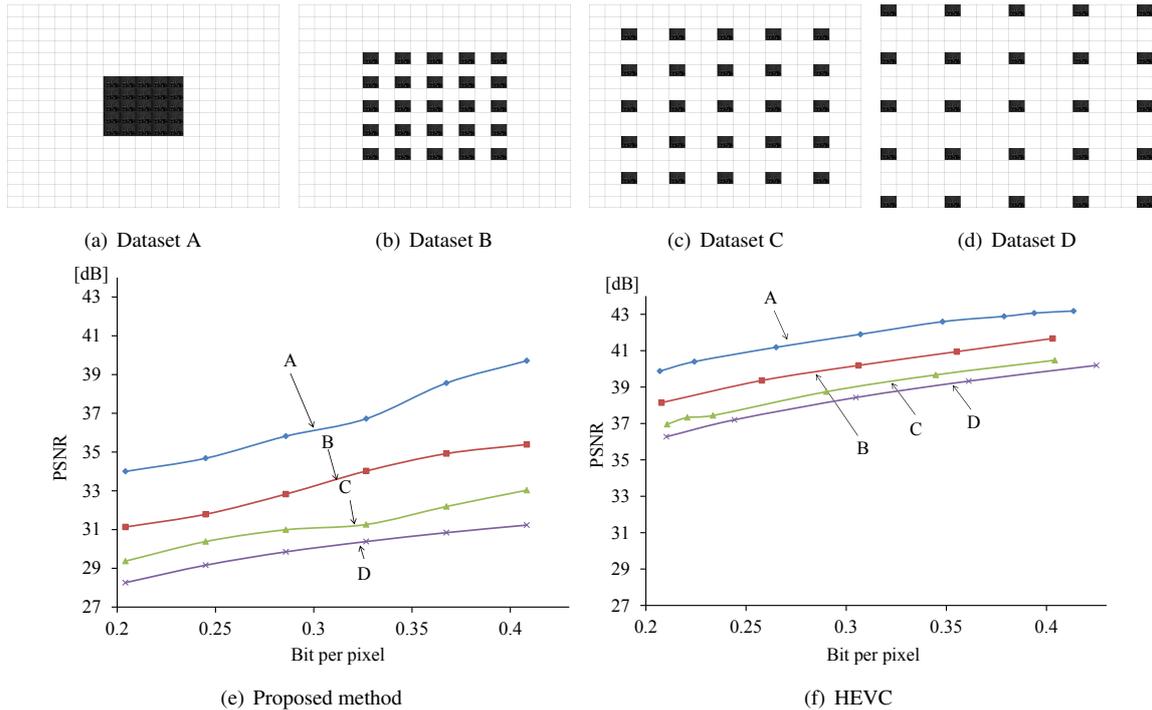


Fig. 10: R-D curves of proposed method and HEVC with 5×5 viewpoints datasets

点画像である。また、図10の(e)(f)を比較するとわかるように、25枚程度の多視点画像では提案手法が優位にはならない。したがって、提案手法の有効性が明確になるのは視点数が十分に多い場合(図7)である。

4 おわりに

本稿では、アクティブシャッター式メガネのために考案された手法を応用し、少数枚の共通の2値画像と画像ごとの重みによって多視点画像を表現する新しい圧縮方法を提案した。提案手法のエンコード処理では2値画像と重みを交互に最適化するが、適切な近似解を簡易に求めて最適化計算の初期値に設定することで、全体のエンコード処理を高速化できることを示した。また、カラー画像に対して提案手法を適用する場合、2値画像をRGBチャンネル間で共通化することで、RGBチャンネルを独立に扱う場合より高いレート歪み特性を達成できることを示した。さらに、提案手法と従来の映像符号化手法の圧縮性能を比較した結果、提案手法では、H.264やHEVCと同等以上のレート歪み特性と、それらを上回る高速なデコードを達成できることを確認した。

今後の展望として、重みの空間的冗長性の活用が考えられる。図2に示したように、重みを画像化したマップ $r_n(k, l)$ は空間的に滑らかである。したがって、変換

符号化を適用することで重みをさらに効率的に圧縮できる可能性がある。また、浮動小数点の座標 (k, l) について重みの値を補間によって求めることもできる。座標 (k, l) は視点位置を表すので、この補間は元のデータに存在しない新たな視点を作りだすことに相当する。すなわち、提案手法では受信側で視点数を増やすことも可能であり、将来的にはこの能力まで含めてレート歪み特性を議論する必要がある。さらに、提案手法のエンコード処理の高速化にも取り組み、より画素数の多いデータに対して提案手法の有効性を検証したい。

参考文献

- [1] Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M. and Hanrahan, P.: Light field photography with a hand-held plenoptic camera, *Computer Science Technical Report CSTR*, Vol. 2, No. 11, pp. 1–11 (2005).
- [2] Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A. and Tumblin, J.: Dappled Photography: Mask Enhanced Cameras for Heterodyned Light Fields and Coded Aperture Refocusing, *ACM Trans. Graph.*, Vol. 26, No. 3 (2007).

- [3] Nagahara, H., Zhou, C., Watanabe, T., Ishiguro, H. and Nayar, S. K.: *Programmable Aperture Camera Using LCoS*, pp. 337–350, Springer Berlin Heidelberg (2010).
- [4] Bishop, T. E. and Favaro, P.: The Light Field Camera: Extended Depth of Field, Aliasing, and Super-resolution, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 5, pp. 972–986 (2012).
- [5] Cho, D., Lee, M., Kim, S. and Tai, Y. W.: Modeling the Calibration Pipeline of the Lytro Camera for High Quality Light-Field Image Reconstruction, *2013 IEEE International Conference on Computer Vision*, pp. 3280–3287 (2013).
- [6] Marwah, K., Wetzstein, G., Bando, Y. and Raskar, R.: Compressive Light Field Photography Using Overcomplete Dictionaries and Optimized Projections, *ACM Trans. Graph.*, Vol. 32, No. 4, pp. 46:1–46:12 (2013).
- [7] Wetzstein, G., Lanman, D., Hirsch, M. and Raskar, R.: Tensor Displays: Compressive Light Field Synthesis Using Multilayer Displays with Directional Backlighting, *ACM Trans. Graph.*, Vol. 31, No. 4, pp. 80:1–80:11 (2012).
- [8] Wanner, S. and Goldluecke, B.: 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 (2014).
- [9] Johannsen, O., Sulc, A. and Goldluecke, B.: What Sparse Light Field Coding Reveals about Scene Structure, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3262–3270 (2016).
- [10] Suzuki, T., Takahashi, K. and Fujii, T.: Disparity estimation from light fields using sheared EPI analysis, *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1444–1448 (2016).
- [11] Saito, T., Kobayashi, Y., Takahashi, K. and Fujii, T.: 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 (2016).
- [12] Sullivan, G. J., Boyce, J. M., Chen, Y., Ohm, J. R., Segall, C. A. and Vetro, A.: Standardized Extensions of High Efficiency Video Coding (HEVC), *IEEE Journal of Selected Topics in Signal Processing*, Vol. 7, No. 6, pp. 1001–1016 (2013).
- [13] Tech, G., Chen, Y., Müller, K., Ohm, J. R., Vetro, A. and Wang, Y. K.: 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 (2016).
- [14] Koutaki, G.: Binary Continuous Image Decomposition for Multi-view Display, *ACM Trans. Graph.*, Vol. 35, No. 4, pp. 69:1–69:12 (2016).
- [15] <http://lightfield.stanford.edu/lfs.html>: .