

画面内内挿予測符号化における符号化効率改善に向けた一検討 Coding Efficiency Improvement of Interpolative Prediction based-Intra Coding

渡邊 真由子† 北原 正樹† 清水 淳†
Mayuko Watanabe Masaki Kitahara Atsushi Shimizu

1. まえがき

従来、映像符号化技術は MPEG-2, H.264/AVC[1]に見受けられるように符号化効率向上を目的として発展してきたが、同時に符号化・復号演算量の増加という問題を抱えてきた。次世代映像符号化方式として策定された High Efficiency Video Codec[2] (以下, HEVC) においても、符号化効率が改善された一方、処理が複雑化し符号化・復号演算量の増加が問題となっている。特に、端末への映像配信などを考える場合、復号演算量の増加は再生端末の消費電力増加につながることから、その演算量を削減する符号化方式の検討が必要であると考えられる。特に画面内符号化においては、予測効率の悪化を抑えるためのデブロッキングフィルタが符号化及び復号処理に用いられており、復号処理におけるデブロッキングフィルタ演算量が特に割合として大きくなっている。

そこで著者らは、画面内の復号演算量を削減するため、画面を 4 分割して画面内予測と画面間予測を組み合わせる符号化を行う画面内内挿予測符号化 Interpolative Prediction based-Intra Coding (以下, IPIC) を提案し[3], IPIC を HEVC に適用 (以下, IPIC-HEVC) することで復号演算量を削減できることを確認した。しかし, IPIC-HEVC には, HEVC に比べ符号化効率が低下するという問題がある[4]。本稿では, IPIC-HEVC の符号化効率改善手法として予測効率改善手法を提案し, その実験結果を報告する。2 章で従来手法とその問題点, 3 章で提案手法, 4 章で実験条件を説明し, 実験結果及び考察を 5 章, 6 章に記す。

2. 従来手法

2-1. IPIC

従来手法 IPIC について説明する。まず, 図 1 のように 1 画面を 4 つの分割画像 A,B,C,D に分割する。更に, 分割画像 A は分割画像 A の画素を用いて分割画像内符号化を行い, 分割画像 B,C,D は, 同一画面内に存在する符号化済み分割画像を用いて分割画像間符号化を行う。尚, 分割画像間符号化とは, 通常の画面間符号化のような処理を行うものである。IPIC の処理フローの例を図 2 に示す。

本手法では, 分割画像 B, C, D に対して通常の画面内符号化で用いる外挿予測の代わりに内挿予測を用いるため予測効率が向上する。また, 内挿予測による予測誤差の減少によりブロック歪の発生条件を満たさない画素が増えることからデブロッキングフィルタ演算量が削減される。実際, 低ビットレートにおいて, H.264/AVC, HEVC に対して符号化効率を維持しつつ復号演算量の削減が可能となることが確認されている[3][4]。

†日本電信電話株式会社,
NTT メディアインテリジェンス研究所

original picture



separate

Interleaved sub-picture A, B, C, D

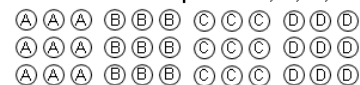


図 1. 分割画像生成法

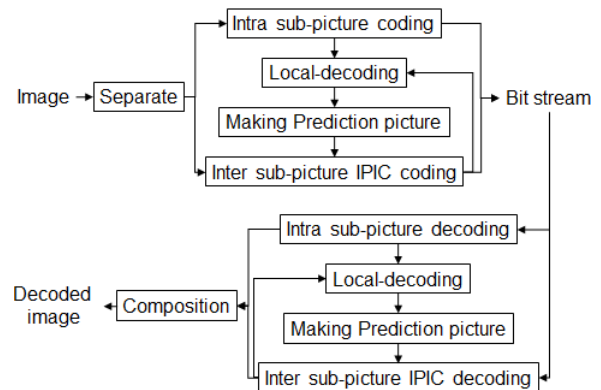


図 2. 画面内内挿予測符号化の例

2-2. IPIC の特徴と問題点

IPIC では, 各分割画像の画素々々に対して原画上での位置関係が定められている。分割画像 B,C の画素は, それぞれ分割画像 A の画素に対して原画上で水平あるいは垂直方向に 1 画素分移動した位置に相当し, また分割画像 D の画素も, B または C の画素を原画上で垂直あるいは水平方向に 1 画素分移動した位置に相当する。このため, ある分割画像の画素位置を整数位置とすると, 他の分割画像の画素は半画素位置に存在する画素となるため, 半画素位置の画素を求める補間フィルタにより予測可能である。

分割画像 B, C, D は, 上述したように符号化済み分割画像に半画素位置の画素値算出フィルタを適用して参照画像を導出し, これを用いて予測誤差符号化を行っている。尚, この手法では画面間予測と同様の符号化を用いているが, 分割画像間の画素の位置関係により各画素に対して適用するフィルタが定まるため, 半画素固定フィルタによる予測が可能である。このため, 通常の画面間符号化であれば符号化が必要な動きベクトルの符号化が不要となり, その符号量を削減できる。尚, 従来検討を行ってきた IPIC-HEVC における半画素固定フィルタには, HEVC の 8 タップフィルタを用いている。この 8 タップフィルタは,

HEVC の半画素位置画素補間に用いられており、IPIC-HEVC において予測が必要な半画素位置の画素予測にも有効であると考えられる。以降の検討では、特に断らない限り半画素フィルタとは HEVC の 8 タップフィルタを意味するものとする。

一方、予測画像の生成にすべて同一の水平方向あるいは垂直方向の半画素フィルタを用いると、フィルタに対して直交するエッジ、または局所的に異なるテクスチャが発生する領域では、周囲の画素による予測が適切でないため予測誤差が増加し、符号化効率が低下する可能性がある。図 3 は、点線のような境界が存在し、符号化効率が低下する例である。図 3 の左図では C の予測画像生成時に A に垂直方向 8 タップフィルタを適用、また右図の場合に B の予測画像生成時に A に水平方向 8 タップフィルタを適用すると、それぞれエッジをまたぐため予測効率が低下する可能性が考えられる。分割画像 D は、B、C 両方から適応的に選択して参照画像として用いることが可能であるが、B、C は従来手法では分割画像 A のみを参照画像とするために、8 タップフィルタによる予測が適切でない領域があるという問題がある。そのため、これを補う仕組みの検討が必要となる。

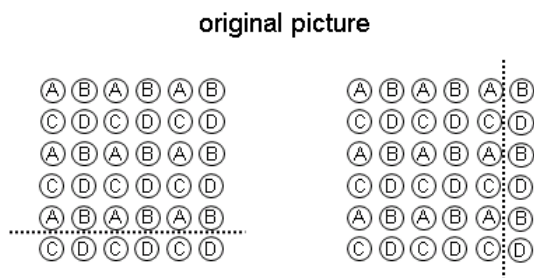


図 3. 予測効率が低下する例

3. 提案手法

以下、本稿で提案する平均値フィルタの適用について説明する。図 4 のように画素が与えられるとき、分割画像 B の画素 b の参照画素の求め方を次のように定める。

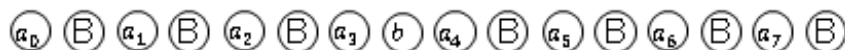
$$|a_3 - a_4| < M1 \text{ のとき}$$

$$\text{フィルタ X. } b = (a_3 + a_4) / 2$$

それ以外

$$\text{フィルタ Y. } b = F(a_i)$$

但し、 $F(a_i)$ は 8 タップフィルタを表すものとし、M は任意の正の整数値である。また、分割画像 C は、分割画像 A の画素との位置関係から、上記を垂直方向に適用することにより算出する。フィルタ Y は従来手法で用いられるフィルタであるため、以下ではフィルタ X について、新たに適用する理由と期待される効果を述べる。



$a_i (i = 0, \dots, 7)$: 分割画像 A の画素
 B, b : 分割画像 B の画素

図 4. 提案手法における画素の予測方法

フィルタ X は、画素 b の両隣の画素の差分が一定値以下のとき適用される平均値フィルタである。これまでの 8 タップフィルタによる予測では、特に直交するエッジの近傍で、エッジの両側の画素を予測に用いるため予測効率が低下する可能性がある。通常は 8 タップフィルタによる予測が良い場合が多いと考えられるが、特に予測対象画素の近傍画素同士の画素値が近い場合には、その予測対象画素も近傍の画素と特に相関が高く、本稿で提案する平均値フィルタ X を適用しても予測は可能である。さらに、仮にエッジが近傍にある場合、エッジの両側の画素を用いずに予測可能となるため、予測効率が向上する。このフィルタ X の適用により、従来手法では B に対して水平方向、C に対して垂直方向の 8 タップフィルタによる予測のみを用いていたが、近傍にエッジがあり 8 タップフィルタが適切でないという場合にも適切な予測を行うことが可能となる。

4. 実験条件

従来の IPIC-HEVC と提案手法による符号化効率、主観画質および復号演算量の比較を行った。

実験には、HEVC Test Model (以下、HM) ver8.0 をベースとして用いた。分割画像 A には HEVC 画面内符号化を適用し、残りの分割画像 B、C、D には、HEVC 画面間符号化の手法を適用した。ここで、分割画像 B、C については、画素単位で提案手法に基づく固定フィルタによる予測画像生成を行っている。尚、従来の画面間符号化で符号化が必要な動きベクトルの情報は、本手法では画素から導出可能であるため符号化していない。フィルタ Y に用いる半画素固定フィルタには、HEVC の半画素予測フィルタとして用いられる 8 タップフィルタを適用した。分割画像 D は、分割画像 A、B と A、C の誤差各々を用いて水平・垂直方向の相関の強さを推定し、D の参照画像を B または C から予測することにより符号化した。また、デブロッキングフィルタについては、従来手法と同様復号演算量削減の効果を維持するため、分割画像毎に処理を行っている。画像サイズは、4 分割画像を用いるために 1920×1080 の下側のラインを間引き、1920×1024 にサイズを変更した画像を実験画像として用いた。

- *実験画像：HEVC 標準化用画像 (Kimono, BasketballDrive, BQTerrace, ParkScene, Cactus)
- *画像サイズ：1920×1024
- *実験枚数：1 枚
- *デブロッキングフィルタ：ON (分割画像単位)
- *固定 QP：QP(A) = 22,27,32,37
 QP(B) = QP(C) = QP(A) + 1, QP(D) = QP(A) + 2
- *閾値 M：
 条件 1. M=2,3,5,7,10,12
 条件 2. M=0

閾値 M は、エンコード時とデコード時に同じ値を用いる必要があるが、それぞれ設定値を入力することとし符号化は行っていない。

ここで、提案手法における参照画素の導出にあたりフィルタ Y を適用する場合は従来手法と同等であることから、条件 2 の $M=0$ を適用したものを従来手法の結果として用いている。

5. 実験結果

以下、符号化効率、主観画質、復号演算量の結果について述べ、各々の考察は次の 6 章で述べる。

5-1. 符号化効率

表 1 は、各画像についての条件 2 に対する条件 1 の BD-Rate[5]である。 $M>0$ のときは、BD-Rate が負値となっており、従来手法に対する符号化効率向上が確認された。また、 M の値を大きくすると、符号化効率が向上する傾向があるが、画像毎に特定の値を超えると、符号化効率の向上する割合は下がることも表 1 から確認される。

表 2 は、各画像、各 QP に対して最適な閾値 M を設定した場合の、 M の値と BD-Rate である。どの画像についても BD-Rate が 0.5% 以上向上することが確認された。

5-2. 主観画質

図 5~9 は、BQTerrace と Cactus の QP=27 のときの条件 1 ($M=10$, $M=7$) と条件 2 のデコード画像である。それぞれ、図 5、7 が従来手法、図 6、8、9 が提案手法により符号化した画像である。図 5 と図 6 は、提案手法により主観画質が向上した領域である。従来手法では上のトランプのエッジがぼんやりしているのに対し、提案手法では同じトランプのエッジがより正確に再現されている領域があることが確認される。

図 7 と図 8、9 は、提案手法により主観画質が悪化した領域の比較である。提案手法で $M=10$ とした図 8 では図 7 に存在しない白いエッジが水平方向にうっすらと現れていることが確認される。しかし、 $M=7$ の結果である図 9 ではエッジがほとんど見受けられない。

5-3. 復号演算量

表 5 は条件 2 に対する条件 1 ($M=10$) の、復号演算量の比較である。これより、すべての画像について復号演算量が削減されることが確認された。

6. 考察

6-1. 符号化効率

表 1 より、 $M>0$ のときは符号化効率が向上していることがわかる。尚、今回の $M \leq 12$ の範囲では、符号化効率は全て従来よりも良い結果を示したが、この M をさらに大きくすると、8 タップフィルタの代わりに平均値フィルタを適用する箇所が増大するため符号化効率は悪くなる。

表 2 の結果より、最適な M の値は画像だけでなく QP によっても変動することがわかる。特に、QP により最適な M が変動する Kimono、Cactus について述べる。

まず、Cactus は、QP が大きくなるにつれ M の値が減少していく傾向にある。Cactus は、今回の実験画像の中では

表 1. 各 M に対する BD-Rate[%]

	M					
	2	3	5	7	10	12
Kimono	-0.40	-0.44	-0.55	-0.59	-0.52	-0.35
ParkScene	-0.16	-0.22	-0.37	-0.50	-0.58	-0.52
Cactus	-0.22	-0.29	-0.33	-0.49	-0.57	-0.57
Basketball Drive	-0.4	-0.94	-0.94	-1.18	-1.58	-1.62
BQTerrace	0.00	-0.06	-0.26	-0.48	-0.85	-1.09

表 2. 最適な設定における BD-Rate[%]

	各 QP の M				BD-Rate
	22	27	32	37	
Kimono	5	5	10	12	-0.67
ParkScene	10	10	10	10	-0.58
Cactus	12	10	7	3	-0.61
BasketballDrive	12	12	12	12	-1.62
BQTerrace	10	10	10	10	-1.09

、特にエッジが多い画像であるが、QP が大きくなるにつれ、本来エッジとなっている領域の画素が平滑化されていくと考えられる。そのため、今回の提案手法では隣接画素同士の差分が閾値以下である場合を、エッジ以外の領域とみなして平均値フィルタを適用しようとしたにもかかわらず、本来はエッジである領域にまで平均値フィルタが適用されてしまう可能性が高くなり、それを抑えるために QP が大きくなるほど M の最適値が減少傾向にあるものと考えられる。

一方、Kimono は QP が小さくなるにつれ、 M が減少していく傾向にある。表 3 の発生符号量をみると、特に Kimono の値が小さいことからわかるように、Kimono はテクスチャの移り変わりやエッジの発生が少ない画像である。そのため、本手法が対象とする領域の境界における画素値の変動による予測効率の低下がおきにくい、QP が小さくなるほど必要以上に平均値フィルタがかかりやすくなってしまい、予測画像から原画上の高周波成分まで削減してしまうため、QP が小さくなるほど適切な M の値が小さくなったと考えられる。

更に、表 1 にて確認された符号化効率向上が、係数符号量の削減によるものであるか確認を行った。表 3 は、QP(A)=27 における分割画像 B,C,D の係数の符号量の比較である。表 3 より、すべての分割画像ではないものの、多くの分割画像について、提案手法は従来手法よりも発生符号量が少なくなっていることがわかる。特に、どの画像についても、分割画像 B または C の少なくともどちらかは係数符号量が削減されていることが確認でき、これは本手法で新たに用いた平均値フィルタにより、予測効率が向上したためと考えられる。分割画像 D についても係数符号量が削減されているが、これは参照画像として用いる分割画像 B または C の SN 比が向上したためと考えられる。

さらに、提案手法により最も符号化効率が向上した画像は BasketballDrive であったが、その中でも特に発生符号量

の多い分割画像 C に対して、提案手法で符号量を削減できたことがわかる。その理由として次が考えられる。

BasketballDrive は、横にぶれたような画像であるため水平方向の相関が強く、さらに水平方向のエッジが多い画像であるが、垂直方向にはエッジが比較的少ない。これは、水平方向フィルタによる予測を行う分割画像 B の符号量が、C に比較して非常に少ないことからもうかがえる。

そのため、従来、予測に垂直 8 タップフィルタのみを用いる分割画像 C には予測効率が悪化しやすいという問題があったが、提案手法では分割画像 A の画素間の差分が閾値以下となる領域が多く、加えてその領域に平均値フィルタを適用したことで予測効率が向上したものと考えられる。

また、**Kimono**、**ParkScene**、**Cactus** では本手法の効果が小さかったが、その理由を **Kimono** と他の 2 画像に分けて述べる。まず、**Kimono** は従来手法でも係数符号量が少なく、提案手法でもその削減率が少なかったためと考えられる。また、**ParkScene** および **Cactus** では、**BasketballDrive** と同程度の符号量が発生しているが、表 3 を見ると、特に大きく符号量削減がされていない。これは、2 画像ともテクスチャが細かい領域が多く、分割画像 A の画素について、隣り合う 2 画素の差分が閾値より大きくなる領域が多いため、平均値フィルタが適用されにくいことが原因と考えられる。しかし、閾値をさらに大きくして実験したところ、逆に符号化効率が悪化することがわかった。そのため、これらの画像のさらなる符号化効率改善には、平均値フィルタ以外の予測画像の生成手法の検討が必要になると考えられる。

上記の分割画像 B または C の発生符号量の削減に対して、予測効率が向上したことが起因しているのかを確認するため、各モードの選択率を調べた。表 4 は、**BasketballDrive** についての、QP=27、M=10 のときの分割画像 B、C、D における各モードの分割画像中に占める割合の比較結果である。ここで、**Skip** は符号化済み分割画像にフィルタを適用して生成される画像のみを用い予測誤差符号化を行わないもの、**Inter** は予測誤差符号化を行うもの、**Intra** は符号化済み分割画像を用いず同一分割画像内に閉じた符号化を指す。

表 4 から、提案手法における **Intra** モードの減少と **Inter/Skip** モードの増加がわかる。**Intra** モードの減少は、提案手法による参照画像と符号化対象分割画像との誤差がこれまでよりも小さくなり、**Inter** または **Skip** での予測が可能な領域が増加したことを表しており、このことから平均値フィルタにより予測効率が向上したと推測される。

また、本稿では閾値を手動で設定し、符号化枚数も 1 枚のみで実験を行っているが、表 2 の結果より閾値をフレーム毎に変えることによる符号化効率向上の見込みがあることから、フレーム毎に適応的に閾値を導出する手法の検討が有効であると考えられ、これは今後の検討課題である。また、その閾値の符号化も必要となるため、それにより符号量が若干増加すると考えられるが、これは多くとも 1 フレームに 1 つの閾値程度で十分であると考えられることから、その符号化により符号化効率が大きく低下するものではないと言える。

6-2. 主観画質

以下、主観画質について考察を行う。

表 3. 発生符号量 (QP=27)

	予測モード	条件 1	条件 2
Kimono	B	778	826
	C	71	91
	D	0	65
ParkScene	B	25586	26366
	C	44928	46002
	D	13028	13236
Cactus	B	68659	68850
	C	42767	42695
	D	18949	19389
BasketballDrive	B	4287	4283
	C	43570	47237
	D	780	906
BQTerrace	B	201902	199161
	C	225268	236204
	D	115354	116064

表 4. 各 CU のモード選択率 (**BasketballDrive**) [%]

	予測モード	条件 1	条件 2
B	Skip	94.7	94.3
	Inter	4.8	4.8
	Intra	0.5	0.9
C	Skip	63.0	59.8
	Inter	24.6	24.9
	Intra	12.4	15.3
D	Skip	98.8	95.2
	Inter	1.0	4.7
	Intra	0.2	0.1



図 5. 条件 1, Cactus, QP=27



図 6. 条件 2, Cactus, QP=27, M=10



図 7. 条件 1, BQTerrace, QP=27



図 8. 条件 2, BQTerrace, QP=27, M=10

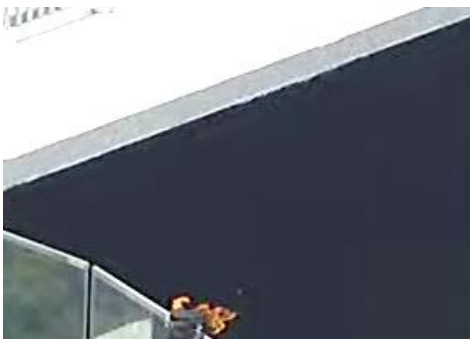


図 9. 条件 2, BQTerrace, QP=27, M=7

まず、図 6 に確認されるようにエッジがきれいに復号された領域について述べる。従来手法では該当領域はエッジ周辺以外の画素に対してもエッジをまたいで 8 タップフィルタによる予測を行ってしまい、その影響で図 5 のようにエッジがきれいに復号されていなかった。しかし、提案手法では、平均値フィルタが適用されたことにより、予測に用いる画素がエッジの片側のみとなったため、歪みが起きにくくなり、視覚上きれいに復号されたと考えられる。

一方、本来存在しないはずのエッジが現れた図 8 であるが、この領域は海面で、全体的に画素間の差分が小さいことから画素同士の差分が全て閾値以下となり、すべて平均値フィルタで予測画素が生成されたためにエッジが現れたと考えられる。しかし図 9 ではエッジが見えなくなっていることから、閾値 M の値を調整することで客観画質、主観画質とも向上させることが可能であると考えられる。

6-3. 復号演算量

最後に復号演算量について記す。

本提案手法は、HEVC 画面内符号化に比べ復号演算量削減率が 10~40%である IPIC-HEVC に対して、復号演算量

表 5. 復号演算量の比較

	QP	条件 1	条件 2	削減率[%]
Kimono	22	0.206	0.227	9.3
	27	0.194	0.217	10.6
	32	0.191	0.209	8.6
	37	0.184	0.206	10.7
ParkScene	22	0.286	0.287	0.3
	27	0.236	0.243	2.9
	32	0.217	0.228	4.8
	37	0.206	0.217	5.1
Cactus	22	0.298	0.299	0.3
	27	0.243	0.251	3.2
	32	0.215	0.234	8.1
	37	0.204	0.221	7.7
Basketball Drive	22	0.234	0.253	7.5
	27	0.21	0.228	7.9
	32	0.203	0.219	7.3
	37	0.19	0.21	9.5
BQTerrace	22	0.318	0.328	3.0
	27	0.273	0.276	1.1
	32	0.236	0.243	2.9
	37	0.213	0.229	7.0

の削減率を維持しつつ符号化効率を向上させることを目的としたものであるが、復号演算量が IPIC-HEVC に比べてさらに削減されることが確認された。

この理由として、次の二つが考えられる。まず、従来技術の予測画像は全て 8 タップフィルタで生成されていたが、今回の手法では平均値フィルタが用いられる領域があり、予測画素を算出するためのタップ長が短くなったために演算量が削減されたことである。二つ目は、8 タップフィルタの代わりに平均値フィルタを適用することで予測効率が良くなる領域があることから、その領域における係数符号量を抑えられたため、係数の復号演算量削減につながったと考えられる。

7. まとめ

本稿では、筆者らが提案する復号演算量の低減のための符号化方式 IPIC について、復号演算量を維持しつつ予測効率を改善する平均値フィルタの適用について検討を行った。その結果、従来検討を行った IPIC-HEVC に対して、0.5~1.6%の符号化効率の改善を確認した。また、その復号演算量については、IPIC-HEVC よりも少なくなることを確認した。

上記の理由として、IPIC-HEVC では予測画像の生成をすべて 8 タップ固定フィルタにより行ったため、予測に用いる画素がエッジをまたぐ場合には予測効率が悪くなる箇所があるという問題に対し、提案手法では隣り合う画素間の差分が一定値以下のときに平均値フィルタを適用することで予測効率を向上したことがあげられる。さらに復号演算量の削減は、8 タップフィルタの代わりとして半画素フィルタの適用、及び予測誤差の削減により実現できたと考えられる。

今回の検討では、従来検討を行ってきた復号演算量削減のための符号化方式 IPIC に対する符号化効率向上の手法を検討し、符号化効率の向上に加え、復号演算量についてもさらに削減できることを確認した。今後の検討では、本稿で用いた閾値を適応的に定める手法に加え、符号化効率向上のための予測手法の改善及び復号演算量のさらなる削減に向けて取り組む予定である。

8. 謝辞

本検討に用いた画像の一部は、NTT ドコモ様の許諾を得て使用したものです。この場を借りてお礼申し上げます。

参考文献

- [1] ITU-T H.264: ITU-T Rec. H.264, “Advanced video coding for generic audio visual services,” 2003.
- [2] JCT-VC, “HM8: High Efficiency Video Coding (HEVC) Test Model 8 (HM 8) Encoder Description,” JCTVC-J1002, JCT-VC Meeting, Stockholm, July. 2012.
- [3] 渡邊真由子, 北原正樹, 清水淳, 如澤裕尚, “画面内符号化における復号演算量削減の検討,” 2011 年信学会総合大会講演論文集, D-11-39, March 2011.
- [4] 渡邊真由子, 北原正樹, 小野尚紀, 清水淳, “復号演算量低減に向けた画面内挿予測符号化への HEVC 適用に関する一考察,” FIT2012, No.3, I-024, pp.301-302, Sept. 2012.
- [5] G. Bjøntegaard, “Calculation of average psnr differences between rd-curves,” ITU-T Q.6/SG16 VCEG, VCEG-M33, Apr. 2001.