RI-004

Poisson 画像合成を用いた動画像符号化方式
Video Coding using Poisson Image Editing

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

1. はじめに

本稿では、Poisson Image Editing[1]（以下、PIE）を用いた画面間符号化を提案する。新たな映像符号化標準として策定された HEVC[2]の高い符号化性能は、新たな予測モードや後処理フィルタ、小数画像生成フィルタの追加・改良等により実現されている。その一方、依然として物体の境界を含む場合の予測には弱いという問題がある。

これは、従来の予測画像は符号化対象ブロックに対して誤差最小となるブロックを選択、もしくはブロック内の画素を一律に小数画素分移動する小数画素フィルタ適用、または前後フレームから選択した 2 つのブロックの重みづけ和算出、などのように基本的に前後のフレームの画素の選択または複数枚フレーム使用時も互いの特性を考慮せず双方を平等に扱い予測画像を生成、のどちらかにより生成されるためである。

さて、筆者らは、画像合成の手法の一つとして知られる PIE の適用による予測画像改善の可能性を検討している[3]。PIE とは、Poisson 方程式という 2 階の楕円型偏微分方程式に由来する画像合成手法である。これによると、2 つの画像が与えられ、一方の画像のテクスチャをもう一方の画像が背景となるように合成する際、各々の画素値とその勾配を Poisson 方程式の境界条件とするときの解を求めることで、自然な形で貼り付けた合成画像を生成できる。この PIE により、一つの符号化対象ブロック内で複数の物体が異なる動きをする場合にも適切な予測画像を生成できることが期待される。

今回、PIE を HEVC 符号化に適用した際の符号化効率を検討し、最大 0.8%向上という結果を得たので報告する。

尚、本稿では、2 章で従来手法とその課題、3 章で提案手法とその基礎となる技術を説明し、4 章では実験の概要、5、6 章で実験の内容と結果、7 章で全体のまとめを述べる。

2. 従来手法とその課題

従来手法である HEVC のインター予測画像生成とその課題を説明する。

インター予測では、参照画像を 1 枚または 2 枚用いて予測画像を生成する。参照画像 1 枚の場合、符号化対象ブロック毎に前後フレームから誤差最小となる同じサイズのブロックを探索する。更に小数画素精度フィルタでブロックを一定方向に小数画素分平行移動させることでより予測精度を向上させることも可能である。

参照画像 2 枚の場合、一定の重みづけ和により予測画像が生成される。この重みづけは符号化ブロック毎に設定され、ブロック内部の局所的な変動には対応できない。このため、符号化対象ブロック内の各テクスチャが小さくかつ各々が異なる動きをする場合、従来手法のように参照画像 2 枚の和では、全体の誤差は最小であるものの各々のテクスチャ部分が平均的に混じり合う画像が作成されるため適切な予測画像が生成できない可能性がある。

† 日本電信電話株式会社，メディアインテリジェンス研究所
NTT Media Intelligence Laboratories, NTT Corporation

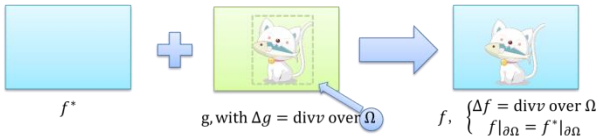

Fig.1 Poisson Image Editing 概略図

3. 提案手法

以下、PIE の概要、提案方式である PIE を用いた符号化方式、PIE による予測画像生成手法の順に説明する。

3.1 PIE

PIE は、ある画像のテクスチャ（前景）と背景となる別の画像を滑らかに合成する画像処理手法の一つであり、Poisson 方程式と呼ばれる次の微分方程式の近似解が合成画像を表す。

$$\begin{cases}\Delta f = \mathrm{div}v \ over \Omega \\ f|_{\partial\Omega} = f^*|_{\partial\Omega}\end{cases} (\exists f^*, v: given) \cdots (1)$$

ここで、Ωは合成する領域、$\partial\Omega$ はその境界を意味する。(1)の第一式は領域Ω上の前景と合成画像のテクスチャ、すなわち勾配の一致を意味し、第二式はΩの境界上における背景と合成画像の画素値の一致を意味する。これらを満たすfを求めることにより、背景がf^*、テクスチャがdiv$v$の自然な合成画像が得られる。

例を、Fig.1 を用いて説明する。ネコのいる画像から点線内を切り取り、背景に合成することを考える。ここでは、左側の無地の画像がf^*、中央のネコの画像はg（このとき、$\mathrm{div}v = \Delta g$）、右側の合成後の画像がf、Ωは点線内の領域、$\partial\Omega$ は点線部分である。このとき、ネコのテクスチャを表現する画素値の勾配を維持し、合成される境界では背景の画素値を保つように合成すると、別の色の背景上に自然にネコが存在するような画像が生成される。尚、以降の説明では、前景を含む画像を勾配画像、背景を含む画像を背景画像と呼ぶ。

3.2 提案方式

提案手法を実現する符号化方式を説明する。

Fig.2 に、符号化時の処理を示す。従来手法との主な相違点は、PIE 予測画像を生成するための勾配画像と背景画像の探索処理、PIE 合成処理及び PIE フラグの追加である。予測画像生成時、まず勾配画像と背景画像をそれぞれ決定し、これらから生成される合成画像と従来のインター予測画像とを比較、誤差が最小となるものを選択する。

ここで合成画像と従来の予測画像との切り替え情報は PIE フラグとして符号化するが、Fig.2 に PIE フラグ出力判

定とあるように、PIE フラグが不要な場合が存在する。これを説明する。(1)式において、次の成立を仮定する。

$$\mathrm{div}v = \Delta f^* \quad over\ \Omega \cdots (2)$$

このとき、Poisson 方程式の求める解fはΩにおいてf^*に一致、つまり合成処理が不要となり、従来の予測画像を用いれば良いことがわかる。特に勾配画像と背景画像が一致する場合、上の仮定を満たすため合成処理不要となり、勾配画像と背景画像が一致するか否かを先に判定可能かつ両者が一致する場合には、PIE フラグ不要かつ従来の予測画像を用いれば良いとわかる。

上述の通り、PIE 合成画像生成には、Fig.3 のように勾配画像gと背景画像f^*が必要である。しかし、勾配画像と背景画像双方の動きベクトル情報を単純に符号化するとベクトルの情報量が 2 倍となる。そのため本方式では、背景画像はテンプレートマッチングを用いて動き探索し、勾配画像の動きベクトル（以下、PIE ベクトル）のみをシンタクスとすることでオーバーヘッド符号量増加を抑える。

次に、Fig.4 に復号時の処理を示す。符号化処理で説明した通り、PIE フラグが存在しない場合があることに注意が必要である。PIE フラグの存在確認のため、テンプレートマッチングの結果と復号されたベクトルの指す領域が一致するかを確認し、両者が同一の場合、PIE フラグは存在しないとして、復号されたベクトルを従来の動きベクトルとみなして復号処理を行う。逆に同一でない場合、PIE フラグを復号し、PIE 処理有ならば復号されたベクトルを PIE ベクトルとして、PIE 処理無ならば復号されたベクトルを動きベクトルとして扱う。

従来の予測画像生成への PIE 適用により、物体境界を含む符号化ブロックにおける予測画像劣化の問題が解決され、予測画像の誤差削減が期待される。これは、符号化ブロック内に複数物体があり、参照画像探索時にその全ての物体に対して適切な画像が参照画像リスト上に存在しない場合、新たに合成画像として予測画像が作成可能となるためである。

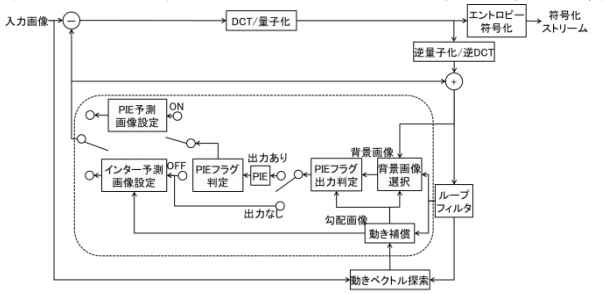

Fig.2 提案符号化方式

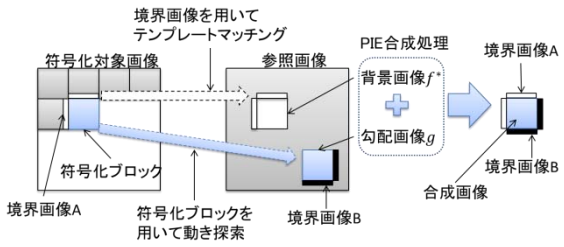

Fig.3 勾配画像・背景画像・境界画像

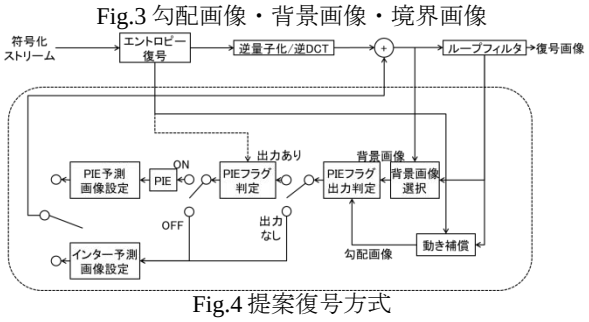

Fig.4 提案復号方式

3.3 PIE による予測画像生成手法

勾配画像gと背景画像f^*は、共に参照画像から決定されるが、その探索手法を説明する。勾配画像は、従来の動きベクトルの探索中心を中心として従来の動き探索により決定される予測画像を設定する。これは、従来の動き探索は歪最小の領域を決定するため、決定される予測画像の勾配が原画像の勾配とも相関が高いと考えられるためである。

背景画像f^*の探索には、先述の通りテンプレートマッチングを用いる。本稿では、テンプレートマッチングの探索中心として PIE ベクトルの探索結果を用いることとした。これは、より適切な背景画像を、探索範囲を限定して演算量の増大を抑えつつ見つけるためである。テンプレートマッチングには符号化対象ブロックに隣接する符号化済み画素（以下、境界画像）を用い、勾配画像と同じフレーム内で境界画像上の誤差が最小となる領域を探索し、これに囲まれる領域を背景画像f^*として設定する。これは、Fig.3 の境界画像 A を用いた探索である。背景画像は、この求め方より左上方向に関する相関が高い。ここで、一般に背景画像と勾配画像の Poisson 画像合成時にはそれらの周囲の画素も用いる必要があるが、この画素として左上の帯は符号化済み画素（Fig.3 の境界画像 A）、右下は勾配画像に隣接する画素（Fig.3 の境界画像 B）を用いることで、背景画像の右下画素のずれによる合成画像としての原画からの誤差拡大を抑制する。

尚、本稿では、PIE の計算方法は PU サイズによらず同一であるが、合成された画像の誤差の精度は PU サイズにより影響を受ける可能性がある。

4. 実験概要

HEVC 符号化方式への PIE 適用について、HEVC テストモデル HM11[4]を用いて画面間符号化の予測画像生成処理部に提案手法を実装し、また HEVC の予測単位の PU サイズによる PIE 処理の効果への影響を合わせて検討した。本稿で行った実験は大きく分けて次の 2 つである。

（1） HEVC に対する符号化効率の評価
（2） PIE 処理の PU サイズ制限時の符号化効率の評価

目的は次の通りである。（1）は、本提案手法の符号化効率に関する検討、また PIE が有効となる PU サイズを検討することを目的とし、これをもとに（2）で PU サイズに応じて PIE 処理を制限した際の符号化効率への影響を検討する。各々の実験内容と結果・考察は 5、6 章の実験 1、実験 2 で述べるが、先に共通の実験条件を説明しておく。

本検討では画面間符号化時のインターPU に対して PIE を適用する。PIE による合成画像と従来の予測画像との切り替えを、PU 毎のシンタクスとして追加した PIE フラグにより実現する。従来の動きベクトルのシンタクスは、PIE が ON の場合は PIE ベクトルを意味し、OFF の場合は従来の動きベクトルを意味する。また、PIE の合成処理には、ガウス・ザイデル法[5]を用いた。

・比較対象：HM11（HEVC テストモデル）
・設定条件：lowdelay_P_main
(1 枚目は I ピクチャ、2 枚目以降は P ピクチャ)
・フレーム枚数：21 枚（PIE 対象フレーム：20 枚）
・固定 QP：22,27,32,37
・実験画像（フレームレート）（解像度 1920x1080）：
BasketballDrive（50）
Kimono（24）
park_joy, crowd_run[6]（30）
・動きベクトル探索範囲：64x64

PIE の条件は、次のパラメータの通りである。今回の実装では、これらの追加パラメータはエンコーダ・デコーダで共通とし、符号化シンタクスには組み込んでいない。

・境界画像の帯幅（Fig.3 の境界画像の帯幅）：2 画素分
・境界画像探索範囲：15x15
・ガウス・ザイデル法の反復演算回数：15 回

5. 実験 1

5.1 実験内容

先に述べた通り、HM に提案手法を実装し、HM に対する符号化効率、PIE フラグの発生符号量と PIE フラグを除外した際の符号化効率、PU サイズ毎の PIE 処理有効となる比率、更に PIE 処理が有効となる PU サイズ毎の PIE フラグの個数について調べた。実験条件は上の通りである。

5.2 実験結果・考察

Table1 は、HM と提案手法の Bitrate、Y-PSNR 及び、HM に対する提案手法の BD-Rate[7]を算出したものである。

BD-Rate を見ると、park_joy、crowd_run では負値となり符号化効率が向上する一方で、BasketballDrive、Kimono では正値となり、符号化効率悪化が確認された。符号化効率が向上した画像は、各々背景や人物が細かいテクスチャを持ち、更にそれらの動きも多様であるため、PIE の効果が表れたものと考えられる。

一方、BasketballDrive の背景は体育館の床や壁であまり細かいテクスチャは存在せず、前景となる人物も符号化ブロックのサイズに対して大きい。また、Kimono は背景の木々は細かいテクスチャであるが、それらの動きは一定の方向に一律に進むため、動きに多様性がない。両者とも PIE で改善対象となりうる、符号化ブロック内での細かいテクスチャ毎の異なる動きが少ないため PIE の効果が得られにくいと考えられる。更に、BasketballDrive の床や壁、Kimono の木々や葉は一面似通っているため、テンプレートマッチングでの結果と PIE ベクトルの結果に微妙なずれが生じ、結果として PIE 不要であるにも関わらず PIE フラグの発生が抑制されなかったことも一因と考えられる。結果として、PIE の改善効果以上に PIE フラグの発生符号量が多いため、符号化効率が悪化したと推測される。

Table2 は、各画像の QP 毎の PIE フラグの発生符号量の 20 フレーム分の合計、及び PIE フラグの発生符号量がないと仮定した場合の符号化効率である。Table1 では、符号化効率が悪化したものもあるが、Table2 では全て 1%以上符号化効率が向上している。このことから、Table1 の結果で、Kimono、BasketballDrive で符号化効率が悪化した理由は、PIE 処理による予測画像改善効果を上回る PIE フラグによるオーバーヘッド符号量の増加にあると考えられる。また、

Table1.HM に対する符号化効率

		HM		Proposed		
Sequence	QP	Bitrate [kbit/sec]	Y-PSNR [dB]	Bitrate [kbit/sec]	Y-PSNR [dB]	BD-Rate[%]
Kimono	22	7830.36	42.01	7824.19	42.01	+0.18
	27	3846.03	40.06	3837.97	40.05	
	32	1935.66	37.36	1934.11	37.35	
	37	984.72	34.64	988.37	34.65	
Basketball Drive	22	15476.11	39.92	15524.57	39.92	+0.15
	27	5476.00	38.50	5472.61	38.49	
	32	2685.52	36.86	2687.60	36.85	
	37	1430.70	34.90	1435.01	34.90	
crowd_run	22	31871.44	39.08	31693.50	39.10	-0.8
	27	14753.81	35.53	14658.99	35.55	
	32	7209.07	32.29	7204.81	32.31	
	37	3642.79	29.45	3651.71	29.47	
park_joy	22	27158.10	40.56	27083.27	40.57	-0.71
	27	12954.75	36.91	12883.66	36.92	
	32	6004.74	33.46	5959.74	33.47	
	37	2792.31	30.64	2783.05	30.65	

Table2. QP 毎の PIE フラグ発生符号量[bit]と PIE フラグを除外した際の符号化効率[%]

	PIE フラグ発生符号量[bit]				BD-Rate[%]
	QP				
Sequence	22	27	32	37	
Kimono	41936	35085	24680	14530	-1.2
BasketballDrive	34676	21934	15722	10263	-1.1
crowd_run	250236	192699	124059	69920	-2.9
park_joy	129231	98881	63023	15898	-1.9

Table3. PU サイズにおける PIE 処理有効な PU 数比率[%]

PU サイズ	Kimono	Basketball Drive	crowd_run	park_joy	Average
8x8	2.14	4.53	8.20	6.01	5.22
4x8	38.89	24.51	30.52	31.16	31.27
8x4	36.46	24.25	31.99	30.99	30.92
16x16	1.38	2.71	3.32	3.12	2.63
16x8	28.97	17.83	18.78	19.19	21.19
8x16	28.99	19.02	16.33	22.04	21.59
16x12	23.95	14.27	13.05	13.54	16.21
12x16	27.46	18.95	11.56	16.75	18.68
16x4	40.63	27.08	27.89	29.35	31.24
4x16	37.83	28.64	28.14	30.93	31.38
32x32	1.12	1.64	0.63	0.91	1.07
32x24	6.63	10.51	2.55	4.64	6.08
24x32	8.07	9.95	2.72	8.33	7.26
32x16	10.76	13.21	5.49	5.73	8.80
16x32	11.97	11.76	3.47	10.28	9.37
32x8	21.09	18.00	11.11	12.45	15.66
8x32	24.09	19.01	8.89	22.18	18.54
64x64	0.40	2.15	0.00	0.56	0.78
64x32	1.55	10.03	1.00	2.33	3.73
32x64	2.35	9.26	0.56	5.25	4.36

Table4. PIE 処理有効となる PU サイズ毎の PIE フラグ個数と面積の 4 画像の平均値（1 フレーム平均）

(A)PU サイズ	(B)PIE フラグ数［個］	(C)面積 ((A)×(B))
8x8	70.225	4494.4
16x16	9.640625	2468.0
32x32	2.015625	2064.0
64x64	0.41875	1715.2

Table1 の Bitrate と Table2 の PIE フラグ発生符号量の QP 毎の推移から明らかなように、PIE フラグ発生符号量の全体の符号量に占める割合は、QP 増加につれて増大することから、本方式は低ビットレートよりも高ビットレートで効果が高いことが期待される。尚、Table2 の PIE フラグの発生符号量除外時の BD-Rate は、厳密には各モードや符号化ブロックサイズの決定時に用いるコスト計算に、PIE フラグの符号量を含めるため、実際に PIE フラグが発生しない場合、若干のずれが生じることを注意として述べておく。

次に Table3 は、各 PU サイズにおける PIE 処理が有効となる PU 数の比率である。これは、画像全体での PIE 処理有効となる PU 数の割合ではなく、各 PU サイズにおいてどの程度 PIE 処理が有効であるかを示すものである。これからわかるように、PU の形状が正方形ではなく横長あるいは縦長の長方形の場合に PIE 処理が有効となりやすい傾向がある。長方形形状の PU で PIE が選択されやすいのは、背景画像のテンプレートマッチングにおいて相関が低くなる右下方向の画素と、境界画像の上または左側の画素との距離が、正方形形状の PU に比べて短く、動き探索が当たりやすいためと考えられる。

ここで、PIE 処理有効となる比率が低い正方形 PU について、PIE 処理有効となる PU サイズ毎の個数と面積の平均値を Table4 に示す。Table4 より、16x16PU、32x32PU、64x64PU では PIE 処理有効となる PU 数が 1 フレームあたり 10 個未満であるとわかり、これらの PU サイズにおける PIE 処理削除に対する予測画像改善への影響は少ないと考えられる。但し、64x64PU は、PIE フラグひとつに影響を受ける画素数が 64x64=4096（画素数）と他のサイズの PU に比べ大きな範囲に及ぶこと、また Table3 における PIE 有効比率と PIE フラグ数からも計算できるように、64x64PU そのものの個数が少なく、PIE フラグ符号量削減効果も少ないことから、PIE フラグ除外は効果が低いと考えられる。

これらより、PIE による予測画像の予測誤差削減効果と PIE フラグによる発生符号量への影響を考慮すると、16x16PU 及び 32x32PU で PIE 処理を除外することにより符号化効率を向上できると考えられる。

6. 実験 2

6.1 実験内容

実験 1 の結果より、提案手法の実装に対して 32x32PU、16x16PU における PIE 処理を除外、及び PIE フラグ情報の符号化処理を削除し、このときの符号化効率を調べた。

6.2 実験結果・考察

Table5 に符号化結果を示す。全画像で Table1 の結果を上回ることが確認された。特に、実験 1 では符号化効率が悪化した BasketballDrive、kimono についても、PU サイズ制限により HM に対する符号化効率向上を確認した。これより、PIE フラグの適切な PU のみへの適用により、更なる符号化効率向上が可能であると考えられる。

Table5. PIE 処理有効となる PU サイズ制限時の HM に対する符号化効率

		HM		Proposed		
Sequence	QP	Bitrate [kbit/sec]	Y-PSNR [dB]	Bitrate [kbit/sec]	Y-PSNR [dB]	BD-Rate[%]
Kimono	22	7830.36	42.01	7812.37	42.00	-0.06
	27	3846.03	40.06	3822.54	40.05	
	32	1935.66	37.36	1930.05	37.35	
	37	984.72	34.64	990.08	34.66	
Basketball Drive	22	15476.11	39.92	15482.78	39.92	-0.01
	27	5476.00	38.50	5469.90	38.50	
	32	2685.52	36.86	2681.56	36.85	
	37	1430.70	34.90	1431.33	34.90	
crowd_run	22	31871.44	39.08	31693.50	39.10	-0.80
	27	14753.81	35.53	14658.99	35.55	
	32	7209.07	32.29	7204.81	32.31	
	37	3642.79	29.45	3651.71	29.47	
park_joy	22	27158.10	40.56	27073.53	40.57	-0.75
	27	12954.75	36.91	12882.72	36.92	
	32	6004.74	33.46	5963.86	33.47	
	37	2792.31	30.64	2780.17	30.65	

7. おわりに

本稿では、PIE を HEVC 符号化に適用した際の符号化効率を検討し、最大 0.8%向上することを確認した。PIE により改善可能な予測画像の性質が限定されることから、提案手法を適用した場合、新たにオーバーヘッドとして追加された PIE フラグによる発生符号量増加に伴い符号化効率が悪化する可能性があるが、PIE が有効となる確率の高い PU への制限など、PIE 適用の範囲限定により符号化効率悪化を抑えることも可能であることが確認された。

今後、PU 毎の PIE 処理有効となる確率に応じたオーバーヘッド削減、及び勾配画像の動き探索と背景画像のテンプレートマッチングに要する処理時間削減の検討が必要と考える。

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ております。この場を借りて御礼申し上げます。

参考文献

[1] P. Perez, M. Gangnet, and A. Blake, “Poisson Image Editing”, Proc. SIGGRAPH’03, pp.313-318, 2003.
[2] Recommendation ITU-T H.265: High efficiency video coding, 2013.
[3] JCT-VC, “HM11: High Efficiency Video Coding (HEVC) Test Model 11 (HM 11) Encoder Description,” JCTVC-M1002, JCT-VC Meeting, Incheon, Apr. 2013.
[4] 渡邊 真由子, 北原 正樹, 清水 淳, “Poisson 画像合成を用いた予測画像生成に関する一検討”, PCSJ/IMPS2014 予稿集, P-1-02, 2014.
[5] 山崎 俊彦, “100 行で書く画像処理最先端 勾配ベースの画像編集： Poisson Image Editing”映像情報メディア学会誌, Vol.64, No.5, pp.729-737, 2010.
[6] https://tech.ebu.ch/docs/hdtv/svt-multiformat-conditions-v10.pdf
[7] G. Bjøntegaard, “Calculation of average psnr differences between rd-curves,” ITU-T Q.6/SG16 VCEG, VCEG-M33, Apr. 2001.