

卒業論文概要書

Summary of Bachelor's Thesis

Date of submission: 02/06/2019

学科名 Department	情報通信	氏名 Name	堀隼也	指導員 Advisor	渡辺 裕 ⑧
研究指導名 Research guidance	オーディオビジュアル 情報処理研究	学籍番号 Student ID number	1w152334-5 ^{CD}		
研究題目 Title	超解像を用いた時系列画像の画質改善 Quality Improvement of Sequential Images Using Super Resolution				

1. まえがき

近年、4Kや8Kをはじめとする、超高精細動画の技術の発展に伴い、出力機器の高解像度化が進んでいる。また、2018年12月にはBS、CS放送において4K、8K実用衛星放送が開始された。このように超高精細のコンテンツが普及する一方で、システムの改変が早急には困難である地上波放送や、過去に製作されたコンテンツを再生する際には依然として低解像のままである。そのため、こうしたコンテンツを高解像のディスプレイで視聴するには、増加した画素を補間する必要がある。この際、線形的に補間すると画面はガウスアンフィルタをかけたようなぼけた画面となる。そこで、超解像と呼ばれる低解像画像で失われていた高周波成分を復元する技術が近年注目を浴びている。本研究ではこの超解像技術を用いて、符号化劣化によって劣化した時系列画像の画質改善について検討する。

2. 超解像技術

超解像技術は前後フレームに含まれている標本点を対象フレーム補う再構成型超解像と、低解像・高解像画像の2組のデータベース学習させた学習型超解像の二つに分類できる。学習型超解像は単一画像に対し適用できるが、再構成型超解像は時系列画像についてのみ適用できる。図1に示すように、時系列画像において、異なるフレームに映る同一被写体は、画面の相関は高いが、サンプリングされる位置はフレームごとに異なる。再構成型超解像では標本点の異なる信号を集め、位置合わせを行うことで画素を増やし超解像を実現できる。これが再構成型超解像の基本技術である。

3. オプティカルフロー

再構成型超解像では前後フレームと対象フレーム間での被写体の対応する位置をサブピクセル精度で

探索する必要がある。この対応点の探索は、オプティカルフローを予測することで求めることができる。オプティカルフローとは各画素の移動量のことであり、図2に示すように各画素がどの位置に移動したかを表すものである。オプティカルフローは一意的な解を決定することが不可能であるため、オプティカルフローの予測には複雑な計算が必要である。しかし、近年では深層学習の発展に伴い、畳み込みニューラルネットワーク(CNN)を用いてオプティカルフローを求められる。これにより、従来以上の精度でオプティカルフローを予測できるようになり、再構成型超解像の性能の向上が期待される。本手法ではCNNにLiteFlowNet[1]を用いた。

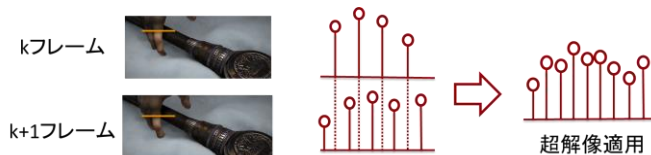


図1 再構成型超解像の理論



図2 オプティカルフローの可視化

4. 提案手法

提案法の概略を図2に示す。動画は符号化によって劣化を起こしてしまう。しかし、対象フレームの信号が劣化によって消失していても、前後のフレームでは正しい信号を有していることがある。そこで、本手法では、オプティカルフローを用いて前後フレ

ームから対象フレームに対応する画素値を計算し、それらの画素値を用いて対象フレームを再構成した予測フレームを作成する。そして対象フレームと予測フレームとの平均フレームを作成することで、対象フレームの符号化劣化を前後フレームから補間することで画質改善を行う。ただし、予測フレーム中に、対応する点が前後フレーム内に存在しない点がオクルージョンとして表れてしまう。そこで、このオクルージョン部分の重みを 0 に設定し、重み平均を計算する。以下の式を満たした画素は符号化画像と大きくずれ、予測が正しくできていないとみなし、重みを 0 とした。

$$| \text{予測画像画素値} - \text{符号化画像画素値} | \geq \text{閾値}$$

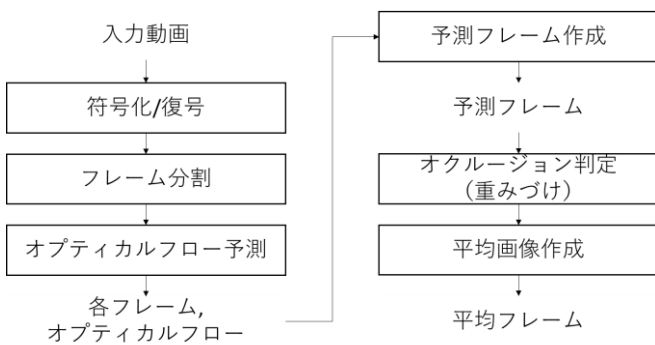


図 3 提案手法のブロック図

5. 実験

MPI Sintel Flow Dataset[1]の 12 種類のテストシーケンスについて、異なる三つのビットレート (500,1000,1500kbps) で符号化し、本手法を適用した。前後 N フレーム (N=1, 2, ..., 9) から平均画像を作成し、参照する予測画像数 N ごとの平均画像の PSNR を測定する。ビットレート 500kbps で符号化したシーケンスに対する結果を表 1 に示す。表中の黄色は元の符号化画像よりも PSNR が向上した結果について色を付けている。また、各ビットレートについて、各シーケンスで最も向上した PSNR の平均値を表 2 に示す、最後に、提案手法による超解像処理結果を図 3 に示す。

得られた画面の PSNR は予測フレーム数 N の大きさによっては、改善されるシーケンスが存在する。高ビットレートでの本手法の適用は低ビットレートほどの効果を得られなかった。これは、符号化劣化が少ないため改善が少ないと考えられる。また、得られた PSNR 向上は十分ではなく、主観評価では画質改善を十分に感じられない、

6. まとめ

本手法では時系列画像から予測画像を作成し、予測画像から平均画像を作成することで、符号化劣化を改善する手法を提案した。シーケンスの特徴により、用いる予測画像数を変化させることで PSNR 向上した。一方で、提案したオクルージョンの判定法では、平均画像は符号化画像に似た画面にしかならず、本手法の効果を十分に活かすことができない。オクルージョンの判別が課題である。

表 1 予測画像数とシーケンスごとの PSNR

Sequence	PSNR[db]									
	Original	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
'ambush_1'	39.134	-0.062	-0.062	-0.07	-0.073	-0.024	0.087	0.136	0.187	0.341
'ambush_3'	37.556	-0.029	-0.079	-0.108	-0.115	-0.059	0.031	0.07	-0.058	0.033
'bamboo_3'	28.461	-0.037	-0.077	-0.146	-0.2	-0.278	-0.348	-0.44	-0.522	-0.633
'cave_3'	32.918	0.046	0.075	0.085	0.096	0.12	0.119	0.097	0.06	0.008
'market_1'	33.72	0.024	0.015	-0.021	-0.058	-0.104	-0.142	-0.187	-0.24	-0.292
'market_4'	30.661	-0.028	-0.048	-0.067	-0.088	-0.114	-0.143	-0.176	-0.215	-0.264
'mountain_2'	42.758	0.065	0.092	0.084	0.072	0.047	0.022	-0.003	-0.029	-0.057
'PERTURBED_market_3'	36.953	0.085	0.133	0.17	0.192	0.206	0.217	0.228	0.245	0.267
'PERTURBED_shaman_1'	40.162	0.08	0.098	0.097	0.072	0.049	0.022	-0.008	-0.045	-0.08
'temple_1'	31.208	0.011	0.003	-0.022	-0.052	-0.096	-0.132	-0.181	-0.222	-0.284
'tiger'	32.854	0.038	0.045	0.044	0.027	0.006	-0.027	-0.062	-0.113	-0.161
'wall'	42.175	0.007	-0.047	-0.101	-0.097	-0.057	-0.019	-0.021	-0.036	-0.01

表 2 各シーケンスで最も向上した PSNR の平均値

bitrate[kbps]	500	1000	1500
PSNR[db]	0.087	0.045	0.029



図 4 生成画像 (ビットレート 500kbps)

参考文献

- [1] T. W. Hui, X. Tang and C. C. Loy: "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation" Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8981-8989, June 2018.
- [2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black: "A Naturalistic Open Source Movie for Optical Flow Evaluation" European Conf. on Computer Vision (ECCV2012), pp. 611-625, Oct. 2012.

2018年度 卒業論文

超解像を用いた時系列画像の画質改善

Quality Improvement of Sequential Images
Using Super Resolution

指導教員 渡辺 裕 教授

早稲田大学 基幹理工学部

情報通信学科

1W152334-5

堀隼也

目次

第1章	序論	1
1.1	研究の背景	1
1.2	本研究の目的	1
1.3	本論文の構成	2
第2章	超解像技術	3
2.1	まえがき	3
2.2	超解像技術の概要	3
2.3	超解像の種類	4
2.3.1	学習型超解像	4
2.3.2	再構成型超解像	4
2.4	オプティカルフロー	4
2.4.1	オプティカルフローの概要	4
2.4.2	オプティカルフローの計算	5
2.5	むすび	6
第3章	提案手法	7
3.1	まえがき	7
3.2	提案手法の概要	7
3.3	処理の詳細	8
3.3.1	符号化	8
3.3.2	予測フレーム作成	8
3.3.3	オプティカルフロー予測	8
3.3.4	予測フレーム作成	8
3.3.5	重みづけ	8
3.3.6	平均フレーム作成	9
3.4	むすび	10
第4章	実験結果・考察	11
4.1	まえがき	11
4.2	作成画面	11
4.3	実験結果	14
第5章	結論	19
5.1	結論	19
5.2	今後の課題	19
	謝辞	20

参考文献.....	21
図一覧.....	22
表一覧.....	23
研究業績.....	24

第1章 序論

1.1 研究の背景

2018年12月より、BS、CS放送において4K、8K実用衛星放送が開始された[1]。4K対応チューナーや4K対応テレビを用意することで、これらの新4K8K衛星放送を受信することができる。また、近年ではインターネット上の動画共有サービスや映像ストリーミング配信サービスにおいても4Kコンテンツの配信がなされており、高精細映像が身近なものとなっている。

4K、8Kは現行のハイビジョンを凌駕する超高解像度の映像である。4K映像は現行ハイビジョンの2K放送の4倍、8K映像は2K放送の16倍の画素数を持ち、より高精細で臨場感のある映像を体験できる。さらに、従来よりも表現可能な色域が拡大、フレームレートが30フレーム/秒から120フレーム/秒に増加、RGB各色8bitの表示は各色10bitへの拡張、より鮮明で美しい映像表現が可能となる。こうした恩恵を受けられる反面、これらの機能を実現するためにコンテンツのデータ量は8K、120フレーム/秒、10bitの場合、256倍にも増加する。そこで、大量のデータを限られた伝送路で送るためには高い圧縮率を実現する必要がある。

動画像の圧縮方式の研究は長きにわたって行われている。Moving Picture Experts GroupとITU-Tが共同で策定したH264/MPEG-4 AVCをはじめとする動画圧縮規格は広く用いられている。そして現在、先に挙げた超高解像度の映像の符号化に対応するための動画圧縮規格を策定すべく、JVETとITU-Tが次世代映像符号化技術の開発を行っている。このように動画圧縮の技術は今後も必要とされており、より効率の良い圧縮方法が模索されている。

一方、近年では深層学習と呼ばれる人間の神経モデルを模したアルゴリズムを用いることで、画像処理や自然言語の分類・認識問題について高い精度が得られている。これはGPUの急速な発展により大量のデータを処理できるようになったことに起因する。深層学習を用いることで、画像処理の分野においても従来の技術では不可能だった精度のパフォーマンスが達成できるようになった[2]。深層学習による画像処理の技術を動画像の圧縮に応用することで、更に効率の良い圧縮の実現が期待されている。

1.2 本研究の目的

低解像度の画像を高解像度のディスプレイに表示するためには画像を拡大表示する処理が必要になる。このとき、出力機器に対して不足する画素数をどのように補間するかが問題となる。例えば、最も近いものを参照し画素を補間する最近傍補間や、周囲4画素から画素を補間する双一時補間のような線形的な補間方法を用いた場合、拡大された画面は焦点の合わ

ないぼやけた画面となる。これは元の画像の画素値をそのまま、或いは平均値を取り、線形的に画素値を求めており、高周波成分が生成されないことに起因する。そこで、高周波成分を復元し、低解像の画像には存在しなかった信号を復元する超解像と呼ばれる技術が研究されている[3]。この超解像の技術を用いることで、伝送において送信側で低解像の画像を送信し、受信側で超解像処理を行い十分な画質の画像を得ることができれば、低ビットレート環境下においても等価的に高解像画像の受信が可能となる。また、従来の方法で受信した画像についても超解像処理を施すことで更なる画質向上を行うことができる。本研究では符号化画像の画質改善を目的とし、画素の移動量であるオプティカルフローを用いて、圧縮符号化や伝送の過程において劣化した画像の画質改善を行う。本稿では超解像技術を用いて、圧縮符号化によって劣化した時系列画像の画質改善について検討する。

1.3 本論文の構成

本論文の構成を以下に示す。

第1章 本章であり、本研究の背景、関連研究及び目的について述べている。

第2章 本研究で用いる超解像技術やオプティカルフローの概要を述べる。

第3章 本研究の提案手法を述べる。

第4章 本研究の実験について及び結果を述べる。

第5章 本研究の考察、結論及び今後の課題を述べる。

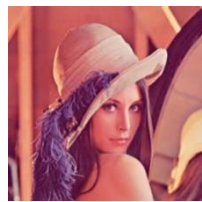
第2章 超解像技術

2.1 まえがき

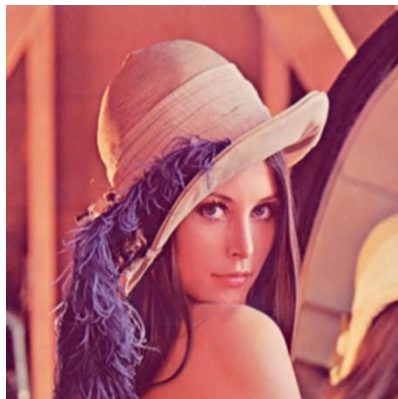
本章では超解像技術の概要と、超解像技術の一つである再構成型超解像について述べる。また、再構成型超解像で利用される、オプティカルフローについても述べる。

2.2 超解像技術の概要

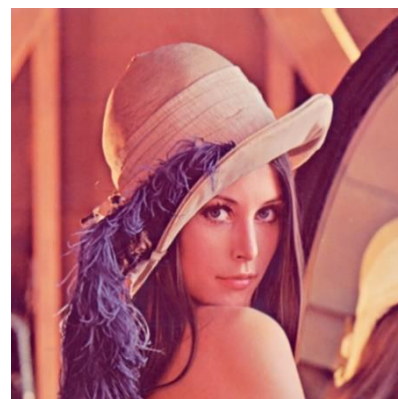
超解像技術は低解像の画像から高周波成分を復元し高解像画像を生成する技術である。周囲の画素を参考にし、輝度を計算する bicubic 法を用いて画像を補間した場合、図 2.1(a) に示すように画像はガウシアンフィルタをかけたようなぼけたものになる。一方で超解像処理を行った場合、図 2.1(b) に示したように明瞭な画面となる。このような超解像技術はデジタルカメラのズーム機能や、監視カメラや衛星写真、医療用光学顕微鏡などの静止画の高画質化、先鋭化に利用されている。また、テレビなどの映像再生機器の一部にも超解像技術は搭載され、映像に対しても超解像技術が利用されている。



オリジナル画像



(a) bicubic 法



(b)waifu2x による超解像

図 2.1 拡大画像の画質比較

2.3 超解像の種類

超解像技術は主に学習型と再構成型の2種類に分類される。学習型は画像単体で超解像を行うことができる技術である。一方で、再構成型では動画などのように、複数枚の連続する画像を利用した超解像技術である。

2.3.1 学習型超解像

学習型超解像は異なる解像度の画像を用意し、高解像画像と低解像画像をそれぞれ小領域のパッチに分割し、対応するパッチを学習させデータベースを作成する。入力画像の各パッチに対応する高解像パッチをデータベースから検索し、低解像パッチを高解像パッチに置き換えるという超解像技術である。

2.3.2 再構成型超解像

再構成型超解像は前後フレームに含まれている標本点を対象フレームに補うことで解像度を向上させる超解像技術である。一般に、連続する画像は変化が少なく画面の相関が高い。一方で、被写体のサンプリングされている位置はフレームごとに変化する。つまり、前後フレームには同一被写体であっても、対象フレームには含まれていない信号が存在する。そこで前後フレームの各画素が対象フレームのどの位置に対応するかをオプティカルフロー(後述: 2.1.3 節) などからサブピクセル精度で位置合わせをし、標本点を増やすことで解像度の向上を図ることができる[4]。

2.4 オプティカルフロー

2.4.1 オプティカルフローの概要

オプティカルフローとはフレーム間の画素の動きをベクトル値で示したものである。図 2.2 のように物体が移動したとき、オプティカルフローは (u, v) と与えられる。また、オプティカルフローを視覚化すると図 2.3 に示すようになる。こうしたオプティカルフローは主に、動画内の物体のトラッキングや、物体の動きのフロー推定などに求められる。また、再構成型超解像において位置合わせを行う際、このオプティカルフローが用いられる。

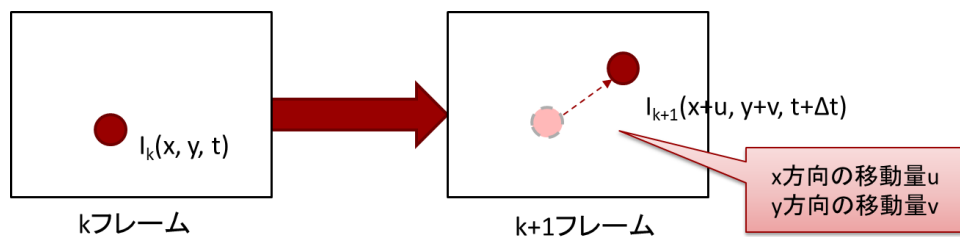


図 2.2 オプティカルフローの概略



図 2.3 オプティカルフローの可視化

2.4.2 オプティカルフローの計算

図 2.2 において，時刻 t である k フレーム上の点 (x, y) の明度を $I(x, y, t)$ と示し，点 (x, y, t) が Δt 秒後， $k+1$ フレーム上，点 $(x+\Delta x, y+\Delta y, t+\Delta t)$ に移動したとする．このとき，オプティカルフローは以下のように求められる[4]．

移動前後で参照する点の明度が不変だと仮定すると，以下の関係が成り立つ．

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (2.1)$$

右辺をテイラー展開し 2 次項以降を切り捨てると以下の式が成り立つ．

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t \quad (2.2)$$

ここで式(1) の関係から，

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0 \quad (2.3)$$

Δt で除算し，

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} = 0 \quad (2.4)$$

ここで, $\frac{\partial I}{\partial x} = I_x$, $\frac{\partial I}{\partial y} = I_y$, $\frac{\partial I}{\partial t} = I_t$, $\frac{\Delta x}{\Delta t} = v_x$, $\frac{\Delta y}{\Delta t} = v_y$ とおく,

$$I_x v_x + I_y v_y + I_t = 0 \quad (2.5)$$

この式を解き, v_x , v_y を求めることで オプティカルフローを求めることができる. しかし, 未知数が v_x , v_y の二つであるのに対し, 方程式は一つしかない. そのため解を一意に求めることができない. そこで, 制約条件として方程式を増やすことで最も正解に近いと推定できるオプティカルフローが求められる. その中でも, Lucas-Kanade 法やブロックマッチング法といった推定法が有名である. このように, オプティカルフローは一意に求められないため, 様々な推定法が研究されている. また, 近年では深層学習の研究が発展しており, オプティカルフローを求める畳み込みニューラルネット (CNN) なども開発されている. CNN を用いたオプティカルフローの算出は従来手法よりも精度や処理速度が向上する. 例えば LiteFlowNet [6]では, 画像間の画素の移動量を計算するのではなく, 画像特徴の移動量を計算し, 従来手法では難しかった, 画像間での変化の大きい部分のオプティカルフローの精度が向上する.

2.5 むすび

本章では超解像技術の概要と, オプティカルフローについて述べた. 再構成型超解像ではオプティカルフローを用いることで超解像を実現できる.

第3章 提案手法

3.1 まえがき

本章では、連続する時系列画像に対して、再構成型超解像を用いて画質改善を行う手法を提案する。3.2節では提案手法の概要を説明し、それ以降の節では提案手法の具体的な説明を行う。

3.2 提案手法の概要

本手法では、圧縮符号化で劣化した画像の画質改善の手法を提案する。その概略図を図3.1に示す。

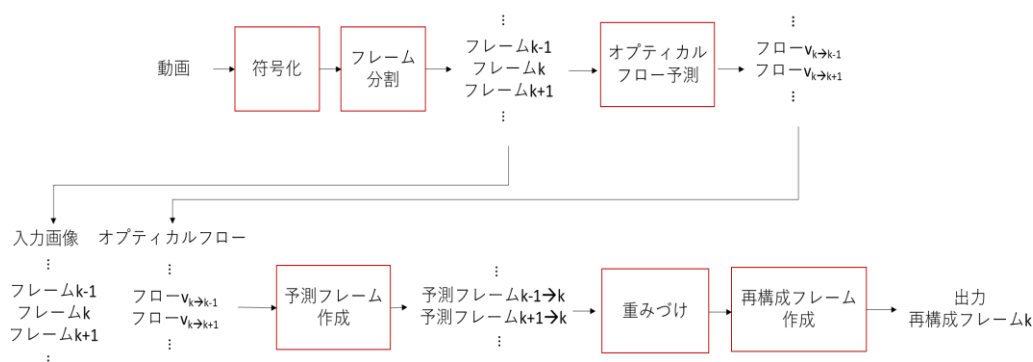


図 3.1 提案する画質改善手法

本提案手法では、時系列画像 I_k, I_{k+1} に対して、各画素について対象画像 I_k から画像 I_{k+1} に対してオプティカルフローを小数精度で計算する。求めたオプティカルフローを用いて、対象フレームの各画素を、前後フレームの画素値から再構成をし、予測フレームを作成する。この際に、オプティカルフローは小数精度で与えられるため、予測フレームの画素値は bicubic 法を用いて複数画素から線形的に求められる。対象フレームと標本点のずれている複数の画素から予測画素を作成するため、予測画素は複数の標本点を集めた画素となる。同様の操作を I_{k+1} だけでなく、 I_k の前後複数枚のフレームに対しても行い、複数枚の予測フレームを作成する。最後に、これら複数枚の予測フレームに対して、オクルージョンの有無で重みづけを行った後、重みに基づいた平均フレームを作成する。異なる標本点を持つ予測画素同士の平均をとることとなるため、画質改善が期待できる。

3.3 処理の詳細

3.3.1 符号化

圧縮符号化した画像を用意するため、入力動画を符号化する。符号化時に、ビットレートを指定し、圧縮の程度を変化させる。また、ビットレートごとの画質の変化を観測するため、H.264において異なるビットレート(500kbps, 1000kbps, 1500kbps)での符号化を行った。これにより、元の画質と画質改善の効果の関係を調査する。

3.3.2 予測フレーム作成

圧縮符号化後、符号化した動画を連番書き出しする。

3.3.3 オプティカルフロー予測

画質改善を行いたいフレームからその前後のフレーム方向のオプティカルフローを予測する。本手法ではオプティカルフローを用いるCNNである LiteFlowNet を用いて、オプティカルフローを求めた。オプティカルフローは小数精度、垂直成分、水平成分で各画素に対して求まる。

3.3.4 予測フレーム作成

予測フレーム作成の概略を図 3.6.1 に示す。k フレームを k+N フレームから再構成する。このとき、k フレームから k+N フレーム方向のオプティカルフローを用いて、k+N フレーム中の k フレームに対応する画素位置を算出する。k フレーム中の座標の画素(x,y)が k+N フレームへ(u,v)だけ動いたとき、k+1 フレーム中の座標(x+u,y+v)の画素に対応することとなる。ここで、オプティカルフロー(u,v)は小数精度で与えられるため、k+N フレーム中の画素値 $I_{k+N}(x+u,y+v)$ を bicubic 法により算出する。この処理を全画素に施すことで予測画面を作成する。

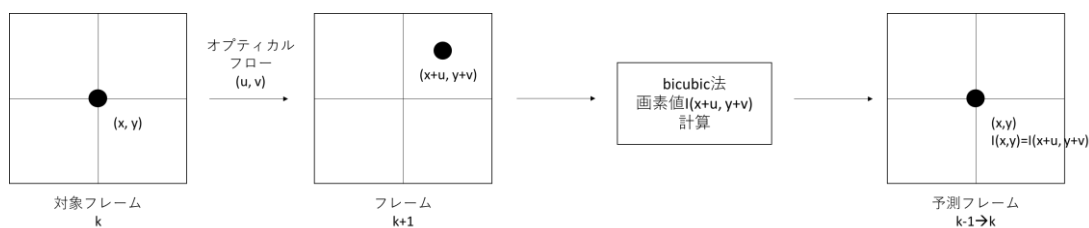


図 3.2 予測画素の作成

3.3.5 重みづけ

k+N フレームから k フレームの予測画面を作成した。この際に、図 3.3 に示したように、オクルージョンと呼ばれる被写体が二重に表示されたり、画面の端が黒く表示されたりする現象が生じる。これは図 3.4 のように、予測先のフレームに存在する部分が参照元のフレ

ームではほかの被写体に遮られ、オプティカルフローの予測が正しくできないために発生する。一般的にオクルージョンは双方向予測を行うことで解消できる。片方では予測できなかった箇所をもう一方の予測によって補足する。本手法では前後数フレームの平均をとることで、標本点を集約したこととなり画質の向上を行う。そのため、各画素がオクルージョンであるかを判定し、オクルージョン部分の重みを 0 にする必要がある。オクルージョンの判定法として、予測画像と符号化画像とで各画素値の差分をとり、その絶対値が閾値以上であれば、その画素をオクルージョンとし、重みを 0 にした。



図 3.3 オクルージョンの実例

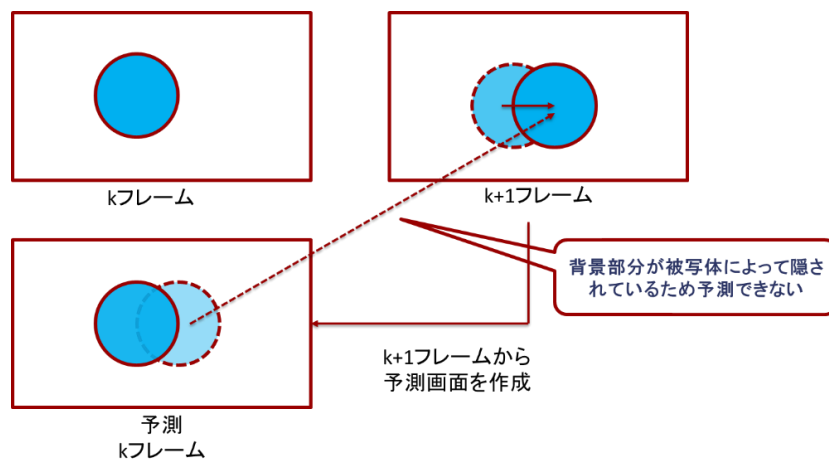


図 3.4 オクルージョンの発生の仕組み

3.3.6 平均フレーム作成

符号化画像と前後 N フレームからの予測画像とで重み平均をとる。 N の値を変化させて PSNR の大きさを測定する。

3.4 むすび

本章では，異なる標本点から集めた画素値の平均を求めることで符号化劣化の改善を行う，提案手法を示した．

第4章 実験結果・考察

4.1 まえがき

本章では提案手法を実行し作成した画面を示し、その後、テストシーケンスにおけるPSNRを測定し、本手法の有効性を確認する。

4.2 作成画面

本実験で作成した画面を図4.1から図4.9に示す。



図 4.1 オリジナル画像(k フレーム)



図 4.2 オリジナル画像(k-1 フレーム)



図 4.3 オリジナル画像(k+1 フレーム)



図 4.4 符号化画像(k フレーム)



図 4.5 符号化画像(k-1 フレーム)



図 4.6 符号化画像($k+1$ フレーム)



図 4.7 $k-1$ フレームから予測した k フレーム画像



図 4.8 $k+1$ フレームから予測した k フレーム画像



図 4.9 前後 1 フレームからの予測画像と k フレームの平均画像

4.3 実験結果

MPI Sintel Flow Dataset[5]の 12 のテストシーケンスについて、本手法を適用した。(ビットレート 500kbps, H.264) オリジナル画像と前後 N フレーム (N=1, 2, ..., 9) から作成した平均画像によるシーケンスの PSNR を表 4.1 に示す。また、得られた画像を図 4.10 に示す。ビットレート 1000kbps, 1500kbps において符号化した画像に対して本手法を適用した。その結果を表 4.2, 表 4.3 に示す。表中の黄色は符号化画像よりも画質が改善したことを示す。

表 4.1 各シーケンスの提案手法の PSNR(500kbps)

Sequence	PSNR[dB]									
	Original	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
'ambush_1'	39.134	39.072	39.072	39.064	39.061	39.11	39.221	39.27	39.321	39.475
'ambush_3'	37.556	37.527	37.477	37.448	37.441	37.497	37.587	37.626	37.614	37.589
'bamboo_3'	28.461	28.424	28.384	28.315	28.261	28.183	28.113	28.021	27.939	27.828
'cave_3'	32.918	32.964	32.993	33.003	33.014	33.038	33.037	33.015	32.978	32.926
'market_1'	33.72	33.744	33.735	33.699	33.662	33.616	33.578	33.533	33.48	33.428
'market_4'	30.661	30.633	30.613	30.594	30.573	30.547	30.518	30.485	30.446	30.397
'mountain_2'	42.758	42.823	42.85	42.842	42.83	42.805	42.78	42.755	42.729	42.701
'PERTURBED_market_3'	36.953	37.038	37.086	37.123	37.145	37.159	37.17	37.181	37.198	37.22
'PERTURBED_shaman_1'	40.162	40.242	40.26	40.259	40.234	40.211	40.184	40.154	40.117	40.082
'temple_1'	31.208	31.219	31.211	31.186	31.156	31.112	31.076	31.027	30.986	30.924
'tiger'	32.854	32.892	32.899	32.898	32.881	32.86	32.827	32.792	32.741	32.693
'wall'	42.175	42.182	42.128	42.074	42.078	42.118	42.156	42.154	42.139	42.165



图 4.10 (a) 实验结果 (左: 符号化画像 右: 提案手法)



图 4.10(b) 实验结果 (左: 符号化画像 右: 提案手法)

表 4.2 各シーケンスの提案手法の PSNR(1000kbps)

Sequence	PSNR[dB]									
	Original	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
'ambush_1'	42.21037	41.901	41.755	41.633	41.592	41.587	41.655	41.704	41.731	41.824
'ambush_2'	40.42889	40.291	40.208	40.156	40.128	40.153	40.248	40.272	40.258	40.225
'bamboo_1'	32.41163	32.398	32.371	32.311	32.26	32.182	32.118	32.023	31.945	31.836
'cave_3'	35.23636	35.288	35.308	35.305	35.302	35.308	35.289	35.243	35.186	35.107
'market_1'	37.65591	37.694	37.676	37.64	37.603	37.559	37.527	37.486	37.452	37.42
'market_4'	34.9182	34.899	34.876	34.857	34.838	34.818	34.796	34.773	34.748	34.716
'mountain_1'	44.70539	44.743	44.744	44.705	44.659	44.611	44.565	44.52	44.48	44.446
'PERTURB_1'	40.3042	40.405	40.436	40.459	40.462	40.459	40.448	40.44	40.448	40.459
'PERTURB_2'	42.33227	42.416	42.424	42.391	42.335	42.286	42.235	42.198	42.155	42.112
'temple_1'	34.93554	34.956	34.946	34.918	34.892	34.844	34.811	34.756	34.714	34.644
'tiger'	36.48962	36.566	36.598	36.611	36.604	36.585	36.555	36.517	36.472	36.426
'wall'	44.01915	43.92	43.802	43.722	43.696	43.701	43.712	43.69	43.656	43.668

表 4.3 各シーケンスの提案手法の PSNR(1500kbps)

Sequence	PSNR[dB]									
	Original	N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8	N=9
'ambush_1'	43.65457	43.103	43.655	42.707	42.62	42.602	42.645	42.628	42.632	42.671
'ambush_2'	41.91891	41.669	41.919	41.473	41.422	41.43	41.5	41.483	41.445	41.407
'bamboo_1'	34.83715	34.843	34.837	34.772	34.733	34.663	34.611	34.527	34.461	34.36
'cave_3'	36.55324	36.583	36.553	36.566	36.545	36.532	36.493	36.427	36.351	36.249
'market_1'	39.70849	39.73	39.708	39.648	39.603	39.552	39.515	39.473	39.441	39.417
'market_4'	37.19442	37.146	37.194	37.08	37.052	37.021	36.99	36.96	36.926	36.892
'mountain_1'	45.41137	45.405	45.411	45.32	45.259	45.189	45.129	45.079	45.029	44.983
'PERTURB_1'	41.91989	41.996	41.92	42.003	41.984	41.96	41.932	41.913	41.91	41.909
'PERTURB_2'	43.33803	43.4	43.338	43.347	43.274	43.206	43.146	43.096	43.039	42.983
'temple_1'	36.9758	36.992	36.976	36.949	36.924	36.877	36.846	36.788	36.749	36.675
'tiger'	38.48622	38.568	38.486	38.62	38.616	38.596	38.563	38.522	38.474	38.42
'wall'	44.7596	44.593	44.76	44.349	44.315	44.306	44.307	44.268	44.236	44.236

得られた画面の PSNR は予測フレーム数 N の大きさによっては改善されるシーケンスが存在する。一方で主観評価では、PSNR の向上が十分ではなく、提案手法により得られた画像は視覚的には変化が読み取れなかった。

また、シーケンス “ambush_1”, ”ambush_3” は画面の動きが大きく、オクルージョンがほかのシーケンスよりも多く発生した。そのため、多くの予測画像の平均効果による画質改善が見られたと考えられる。シーケンス “bamboo” は似た輝度の画素が連続しているため、オクルージョン判定が正常になされず、画質が向上しなかったと考えられる。シーケンス “market1”, “market4” では動きが大きく、対象フレームから離れたフレームでは画面の構図や、被写体が大きく変わるため、予測に利用するフレーム数が多いと画質改善の効果が得られなかったと考えられる。また、シーケンス “temple”, ”wall” においても被写体と同じ動きを繰り返すシーケンスのため、予測に利用するフレーム数が多いと、予測フレームがノイズとなってしまう画質改善の効果が得られなかったと考えられる。一方でその他のシーケンスでは被写体が静止し続けることなく、また、激しく動くこともなく、なだらかな動きであり、本手法の効果が得られた。

また、高ビットレートでの本手法の適用は低ビットレートほどの効果を得られなかった。これは高ビットレートで符号化したためノイズの発生量が少なくなったためと考えられる。

第5章 結論

5.1 結論

本研究では、時系列画像から予測画像を作成し、それらの平均画像を作成することで、符号化劣化を改善する手法を提案した。オプティカルフローから求めた予測フレームは、オクルージョンなどの問題から正確に求められない場合があった。そのため、符号化画像と予測画像の各画素の差をとり、差の大きさによって重みづけを行い、該当する予測フレームの利用の有無を決めた。そのため、符号化で失われた情報を重みづけによって復元することができなかった。平均画像は符号化画像に似た画面にしかならず、本手法の効果を十分に生かすことができなかった。

一方、特定のシーケンスでは、用いる予測画像数を適宜決定するなど、パラメータを変化させることで PSNR の向上が見られた。シーケンスごとに適切なパラメータを設定することで画質改善が期待できる。

5.2 今後の課題

オクルージョン部分の処理を適切に行うことで、対象フレームのノイズの補間を前後フレームから適切に行うことができる。そのため、オクルージョンを含めオプティカルフローをより正確に求める必要がある。

謝辞

本研究のご指導をして頂いた渡辺教授に深謝いたします。

また、本研究にご協力、アドバイスをしてくださった SHARP 株式会社の方々に心から感謝いたします。

参考文献

- [1]. 総務省, 4K・8K放送の推進
http://www.soumu.go.jp/menu_seisaku/ictseisaku/housou_suishin/4k8k_suishin.html
(2019年1月現在)
- [2]. MathWorks, ディープラーニングこれだけは知っておきたい3つのこと
<https://jp.mathworks.com/discovery/deep-learning.html>
(2019年1月現在)
- [3]. 小森秀樹: “超解像技術”, 映像情報メディア学会誌, Vol.63, No.10, pp.1400-1402, Oct. 2009
- [4]. 竹島, 加藤, 金子: “サブピクセルシフト画像を用いた複数フレーム超解像技術”, 東芝レビュー, Vol.64, No.6, pp.11-14 June 2009 .
- [5]. OpenCV, オプティカルフロー(Optical Flow)
http://labs.eecs.tottori-u.ac.jp/sd/Member/oyamada/OpenCV/html/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html
(2019年1月現在)
- [6]. D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black: “A Naturalistic Open Source Movie for Optical Flow Evaluation” European Conf. on Computer Vision (ECCV2012), pp. 611-625, Oct. 2012.
- [7]. T. W. Hui, X. Tang and C. C. Loy: “LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation” Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8981-8989, June 2018.

図一覧

図 2.1	拡大画像の画質比較.....	3
図 2.2	オプティカルフローの概略.....	4
図 2.3	オプティカルフローの可視化.....	5
図 3.1	提案する画質改善手法.....	7
図 3.2	予測画素の作成.....	8
図 3.3	オクルージョンの実例.....	9
図 3.4	オクルージョンの発生原理.....	9
図 4.1	オリジナル画像(k フレーム)	11
図 4.2	オリジナル画像($k-1$ フレーム)	11
図 4.3	オリジナル画像($k+1$ フレーム)	12
図 4.4	符号化画像(k フレーム)	12
図 4.5	符号化画像($k-1$ フレーム)	12
図 4.6	符号化画像($k+1$ フレーム)	13
図 4.7	$k-1$ フレームから予測した k フレーム画像.....	13
図 4.8	$k+1$ フレームから予測した k フレーム画像.....	13
図 4.9	前後 1 フレームからの予測画像と k フレームの平均画像.....	14
図 4.10	実験結果 (左: 符号化画像 右: 提案手法)	15

表一覧

表 4.1 各シーケンスの提案手法の PSNR(500kbps)	14
表 4.2 各シーケンスの提案手法の PSNR(1000kbps)	17
表 4.3 各シーケンスの提案手法の PSNR(1500kbps)	17

研究業績

1. 堀隼也,渡辺裕, ” 超解像を用いた時系列画像の画質改善” ,2019 年電子情報通信学会総合大会 2018 年 3 月 19 日～22 日発表予定