

＜令和2年度卒業研究論文＞

フーリエ観測行列と L_1 ノルム誤差基準を用いた
畳み込み型スパースコーディングに基づく
分散圧縮符号化

指導教員: 黒木祥光 教授

制御情報工学科5年 56237 森崎颯太

目次

1	はじめに	3
2	背景技術	4
2.1	圧縮センシング	4
2.2	ADMM	5
2.3	CSC を用いた分散圧縮符号化	5
2.3.1	畳み込み型スパースコーディング	5
2.3.2	符号化側の処理	6
2.3.3	復号化側の処理	6
2.4	ADMM による畳込み辞書学習	7
2.4.1	辞書 d_m の最適化	7
2.4.2	係数 x_m の最適化	9
2.5	ノンキーフレームに対する係数行列を求める	11
3	提案手法	13
4	実験	14
5	実験結果・考察	15
6	むすび	19
	謝辞	20

参考文献	21
------	----

第1章

はじめに

本研究では畳み込み型スパースコーディング (CSC: Convolutional Sparse Coding) [1] を用いた分散圧縮符号化について検討する。分散圧縮符号化とは、圧縮センシング [2] と分散符号化を組み合わせた符号化方法であり、動画像に適用する場合、フレームをキーフレームとノンキーフレームに分割する。符号化の際、キーフレームは静止画像として符号化し、ノンキーフレームは観測行列を用いて低次元化する。復号側では、復号されたキーフレームからノンキーフレームを復号化するためのサイド情報を生成する。既存の動画像の符号化では動き補償などの処理を符号化側で行うが、分散圧縮符号化での符号化側の処理は与えられた画像を観測行列で低次元化し、それを伝送・蓄積するのみである。そして復号化側の処理にて現画像を推定する処理を行う。このようにして分散圧縮符号化を用いることで符号化側の処理の負荷を軽減することが可能となる。

多くの CSC を用いた分散圧縮符号化の研究では、観測行列にランダム行列を用いるが、本研究ではフーリエ行列を用いることにより、計算量負荷の低減とメモリ容量の削減を試みる。

第2章

背景技術

2.1 圧縮センシング

圧縮センシングとは、低次元の観測信号 $\mathbf{y} \in \mathbb{R}^L$ から原信号 $\mathbf{f} \in \mathbb{R}^N$ を推定する手法である。ここで原信号はスパースな係数 $\mathbf{x} \in \mathbb{R}^M$ と表現行列 $\Psi \in \mathbb{R}^{M \times N}$ の線形結合で表されるという仮定のもとに成り立つ。 \mathbf{y} は \mathbf{f} に観測行列 $\Phi \in \mathbb{R}^{L \times N}$ を乗じることで得る。

$$\mathbf{y} = \Phi \mathbf{f} \quad (2.1)$$

観測行列の N , L の比を観測比 MR (Measurement Rate) とし、以下のように定義する。

$$MR = \frac{L}{N} \quad (2.2)$$

Φ と \mathbf{y} から \mathbf{f} を得るためには、制約条件 $\mathbf{y} = \Phi \Psi \mathbf{x}$ を満たすスパースである \mathbf{x} を推定する。従って、以下の L_0 ノルム最小化問題を求める。

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = \Phi \Psi \mathbf{x} \quad (2.3)$$

しかし、この問題は NP 困難であるため、以下の L_1 ノルム最小化問題に緩和する。

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = \Phi \Psi \mathbf{x} \quad (2.4)$$

2.2 ADMM

ADMM(Alternating Direction Method of Multipliers)[3] とは共に凸関数である f と g の和を最小化するアルゴリズムである.

$$\arg \min_{\mathbf{x}, \mathbf{y}} f(\mathbf{x}) + g(\mathbf{y}) \quad \text{s.t. } A\mathbf{x} - B\mathbf{y} = \mathbf{c} \quad (2.5)$$

式 (2.5) の ADMM の更新式は以下となる.

$$\mathbf{x}^{(j+1)} = \arg \min f(\mathbf{x}) + \frac{\rho}{2} \|A\mathbf{x} + B\mathbf{y}^{(j)} - \mathbf{c} + \mathbf{u}^{(j)}\|_2^2 \quad (2.6)$$

$$\mathbf{y}^{(j+1)} = \arg \min g(\mathbf{y}) + \frac{\rho}{2} \|A\mathbf{x}^{(j+1)} + B\mathbf{y} - \mathbf{c} + \mathbf{u}^{(j)}\|_2^2 \quad (2.7)$$

$$\mathbf{u}^{(j+1)} = \mathbf{u}^{(j)} + A\mathbf{x}^{(j+1)} + B\mathbf{y}^{(j+1)} - \mathbf{c} \quad (2.8)$$

ADMM は式 (2.6) から (2.8) の更新式を反復することで式 (2.5) の解を得る.

2.3 CSC を用いた分散圧縮符号化

2.3.1 畳み込み型スパースコーディング

CSC は画像 $\mathbf{s} \in \mathbb{R}^N$ を辞書フィルタ $\mathbf{d}_m \in \mathbb{R}^K$ とスパースな係数 $\mathbf{x}_m \in \mathbb{R}^N$ との畳み込みの和で表現する.

$$\mathbf{s} \approx \sum_{m=1}^M \mathbf{d}_m * \mathbf{x}_m \quad (2.9)$$

CSC を用いた分散圧縮符号化では復号したキーフレームである \mathbf{f} を用いて辞書を作成する.

2.3.2 符号化側の処理

動画は GOP (Group of Pictures) に分けられる．それぞれの GOP は一つのキーフレームと，複数のノンキーフレームから成る．符号化側の処理はベクトル化されたノンキーフレーム $\mathbf{p} \in \mathbb{R}^N$ に観測行列 $\Phi \in \mathbb{R}^{M \times N}$ を乗じた観測ベクトル $\mathbf{y} \in \mathbb{R}^M$ を生成する．

$$\mathbf{y} = \Phi \mathbf{p} \quad (2.10)$$

2.3.3 復号化側の処理

復号化されたキーフレームである \mathbf{f} を用いて，以下の式を解くことにより辞書を作成する．外れ値に対して頑健な辞書を作成するため誤差項を L1 ノルムで評価する．

$$\arg \min_{\mathbf{d}_m, \mathbf{x}_m} \left\| \sum_{m=1}^M \mathbf{d}_m * \mathbf{x}_m - \mathbf{f} \right\|_1 + \lambda \sum_{m=1}^M \|\mathbf{x}_m\|_1 \quad (2.11)$$

λ は \mathbf{x}_m のスパース性を制御するパラメータである．式 (2.11) において \mathbf{d}_m と \mathbf{x}_m を同時に最適化するのは困難であるため， \mathbf{d}_m と \mathbf{x}_m の片方を固定しながら交互に最適化を行う．

キーフレームとノンキーフレームは時間軸で隣接しているため，物体の動きは非ゼロ係数の位置の変更で表現される．従って，式 (2.11) で求めた \mathbf{d}_m を用いて観測ベクトル \mathbf{y} を基にノンキーフレームの係数を求める．

$$\arg \min_{\mathbf{x}_m} \left\| \Phi \sum_{m=1}^M \mathbf{d}_m * \mathbf{x}_m - \mathbf{y} \right\|_2^2 + \mu \sum_{m=1}^M \|\mathbf{x}_m\|_1 \quad (2.12)$$

μ は \mathbf{x}_m のスパース性を制御するためのパラメータである．

2.4 ADMMによる畳込み辞書学習

2.4.1 辞書 d_m の最適化

辞書 d_m の最適化は以下の式を解くことにより得られる.

$$\arg \min_{d_m} \left\| \sum_{m=1}^M x_m * d_m - f \right\|_1 + \sum_{m=1}^M \iota_{C_{PN}}(d_m) \quad (2.13)$$

制約集合 C_{PN} は以下の形で定義される.

$$C_{PN} = \{x \in \mathbb{R}^N : (I - PP^T)x = 0, \|x\|_2 \leq 1\} \quad (2.14)$$

P はゼロパディング行列でフーリエ領域で計算を行う際に d_m を x_m と同じ次元にするためのものである. ι は指示関数であり, ある集合 A に対して以下の形で定義される.

$$\iota_A(x) = \begin{cases} 0 & x \in A \\ \infty & x \notin A \end{cases} \quad (2.15)$$

ここで $X_m d_m = x_m * d_m$ となる X_m を定義する.

$$X = (X_1, \dots, X_M) \quad (2.16)$$

$$d = \begin{pmatrix} d_1 \\ \vdots \\ d_M \end{pmatrix} \quad (2.17)$$

X と d を式 (2.16) と式 (2.17) とすれば, 式 (2.13) は以下の形に書き換えることができる.

$$\arg \min_d \|Xd - f\|_1 + \iota_{C_{PN}}(d) \quad (2.18)$$

式 (2.18) を ADMM に適用できる形に変形する.

$$\arg \min_d \|\mathbf{g}_0\|_1 + \iota_{\text{CPN}}(\mathbf{g}_1) \quad \text{s.t.} \quad \begin{pmatrix} \mathbf{X} \\ \mathbf{I} \end{pmatrix} \mathbf{d} - \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \quad (2.19)$$

従って、ADMM の更新式は以下となる．

$$\mathbf{d}^{j+1} = \arg \min \left\| \mathbf{X} \mathbf{d} - \mathbf{g}_0^{(j)} - \mathbf{f} + \mathbf{h}_0^{(j)} \right\|_2^2 + \left\| \mathbf{d} - \mathbf{g}_1^{(j)} + \mathbf{h}_1^{(j)} \right\|_2^2 \quad (2.20)$$

$$\mathbf{g}_0^{(j+1)} = \arg \min \|\mathbf{g}_0\|_1 + \frac{\rho}{2} \left\| \mathbf{X} \mathbf{d}^{(j+1)} - \mathbf{g}_0 - \mathbf{f} + \mathbf{h}_0^{(j)} \right\|_2^2 \quad (2.21)$$

$$\mathbf{g}_1^{(j+1)} = \arg \min \iota_{\text{CPN}}(\mathbf{g}_1) + \frac{\rho}{2} \left\| \mathbf{d}^{(j+1)} - \mathbf{g}_1 + \mathbf{h}_1^{(j)} \right\|_2^2 \quad (2.22)$$

$$\mathbf{h}_0^{(j+1)} = \mathbf{h}_0^{(j)} + \mathbf{X} \mathbf{d}^{(j+1)} - \mathbf{g}_0^{(j+1)} - \mathbf{f} \quad (2.23)$$

$$\mathbf{h}_1^{(j+1)} = \mathbf{h}_1^{(j)} + \mathbf{d}^{(j+1)} - \mathbf{g}_1^{(j+1)} \quad (2.24)$$

式 (2.20) を解くと以下のようになる．

$$\mathbf{d}^{(j+1)} = (\mathbf{X}^T \mathbf{X} + \mathbf{I})^{-1} \left\{ \mathbf{X}^T (\mathbf{g}_0^{(j)} + \mathbf{f} - \mathbf{h}_0^{(j)}) + \mathbf{g}_1^{(j)} - \mathbf{h}_1^{(j)} \right\} \quad (2.25)$$

式 (2.25) は計算量を減らすためにフーリエ領域で解く．ここで以下のような行列を定義する． $\hat{\mathbf{x}}_i$ は \mathbf{x}_i のフーリエ変換である．

$$\hat{X}_i = \text{diag}(\hat{\mathbf{x}}_i) \quad (2.26)$$

$$\hat{\mathbf{X}} = (\hat{X}_1, \hat{X}_2, \dots, \hat{X}_M) \quad (2.27)$$

式 (2.25) は以下のように書き換えることができる．

$$\hat{\mathbf{d}}^{(j+1)} = (\hat{\mathbf{X}}^H \hat{\mathbf{X}} + \mathbf{I})^{-1} \left\{ \hat{\mathbf{X}}^H (\hat{\mathbf{g}}_0^{(j)} + \hat{\mathbf{f}} - \hat{\mathbf{h}}_0^{(j)}) + \hat{\mathbf{g}}_1^{(j)} - \hat{\mathbf{h}}_1^{(j)} \right\} \quad (2.28)$$

式 (2.28) の $\hat{\mathbf{X}}^H \hat{\mathbf{X}}$ はサイズが大きく計算量が大きくなってしまふ．そのため、逆行列の補助定理を用いて計算する．

式 (2.21) はソフト閾値処理により求めることができる．従って、以下のように書き換えることができる．

$$\mathbf{g}_0^{(j+1)} = \text{sign}(\mathbf{a}) \odot \max(0, |\mathbf{a}| - \frac{1}{\rho}) \quad (2.29)$$

$$\mathbf{a} = \mathbf{X}\mathbf{d}^{(j+1)} - \mathbf{f} + \mathbf{h}_0^{(j)} \quad (2.30)$$

式 (2.22) の解は以下のような近接作用素により求めることができる.

$$\mathbf{g}_1^{(j+1)} = \begin{cases} \mathbf{P}\mathbf{P}^T(\mathbf{d}^{(j+1)} + \mathbf{h}_1^{(j)}) & \left(\left\| \mathbf{P}\mathbf{P}^T(\mathbf{d}^{(j+1)} + \mathbf{h}_1^{(j)}) \right\|_2 \leq 1 \right) \\ \frac{\mathbf{P}\mathbf{P}^T(\mathbf{d}^{(j+1)} + \mathbf{h}_1^{(j)})}{\left\| \mathbf{P}\mathbf{P}^T(\mathbf{d}^{(j+1)} + \mathbf{h}_1^{(j)}) \right\|_2} & \left(\left\| \mathbf{P}\mathbf{P}^T(\mathbf{d}^{(j+1)} + \mathbf{h}_1^{(j)}) \right\|_2 > 1 \right) \end{cases} \quad (2.31)$$

2.4.2 係数 \mathbf{x}_m の最適化

係数 \mathbf{x}_m の最適化は以下の式を解くことにより得られる.

$$\arg \min_{\mathbf{d}_m} \left\| \sum_{m=1}^M \mathbf{x}_m * \mathbf{d}_m - \mathbf{f} \right\|_1 + \lambda \sum_{m=1}^M \|\mathbf{x}_m\|_1 \quad (2.32)$$

ここで $\mathbf{D}_m \mathbf{x}_m = \mathbf{d}_m * \mathbf{x}_m$ となる \mathbf{D}_m を定義する.

$$\mathbf{D} = (\mathbf{D}_1, \dots, \mathbf{D}_M) \quad (2.33)$$

$$\mathbf{d} = \begin{pmatrix} \mathbf{x}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{x}_M \end{pmatrix} \quad (2.34)$$

\mathbf{D} と \mathbf{x} を式 (2.33) と式 (2.34) とすれば, 式 (2.32) は以下の形に書き換えることができる.

$$\arg \min_{\mathbf{x}} \|\mathbf{D}\mathbf{x} - \mathbf{f}\|_1 + \lambda \|\mathbf{x}\|_1 \quad (2.35)$$

式 (2.35) を ADMM に適用できる形に変形する.

$$\arg \min_{\mathbf{g}} \|\mathbf{g}_0\|_1 + \lambda \|\mathbf{g}_1\|_1 \quad \text{s.t.} \quad \begin{pmatrix} \mathbf{D} \\ \mathbf{I} \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \quad (2.36)$$

従って、ADMM の更新式は以下となる.

$$\mathbf{x}^{j+1} = \arg \min \left\| \mathbf{D}\mathbf{x} - \mathbf{g}_0^{(j)} - \mathbf{f} + \mathbf{h}_0^{(j)} \right\|_2^2 + \left\| \mathbf{x} - \mathbf{g}_1^{(j)} + \mathbf{h}_1^{(j)} \right\|_2^2 \quad (2.37)$$

$$\mathbf{g}_0^{(j+1)} = \arg \min \|\mathbf{g}_0\|_1 + \frac{\rho}{2} \left\| \mathbf{D}\mathbf{x}^{(j+1)} - \mathbf{g}_0 - \mathbf{f} + \mathbf{h}_0^{(j)} \right\|_2^2 \quad (2.38)$$

$$\mathbf{g}_1^{(j+1)} = \arg \min \lambda \|\mathbf{g}_1\|_1 + \frac{\rho}{2} \left\| \mathbf{x}^{(j+1)} - \mathbf{g}_1 + \mathbf{h}_1^{(j)} \right\|_2^2 \quad (2.39)$$

$$\mathbf{h}_0^{(j+1)} = \mathbf{h}_0^{(j)} + \mathbf{D}\mathbf{x}^{(j+1)} - \mathbf{g}_0^{(j+1)} - \mathbf{f} \quad (2.40)$$

$$\mathbf{h}_1^{(j+1)} = \mathbf{h}_1^{(j)} + \mathbf{x}^{(j+1)} - \mathbf{g}_1^{(j+1)} \quad (2.41)$$

式 (2.37) を解くと以下のようになる.

$$\mathbf{x}^{(j+1)} = (\mathbf{D}^T \mathbf{D} + \mathbf{I})^{-1} \left\{ \mathbf{D}^T (\mathbf{g}_0^{(j)} + \mathbf{f} - \mathbf{h}_0^{(j)}) + \mathbf{g}_1^{(j)} - \mathbf{h}_1^{(j)} \right\} \quad (2.42)$$

式 (2.42) は計算量を減らすためにフーリエ領域で解く. ここで以下のような行列を定義する. $\hat{\mathbf{d}}_i$ は \mathbf{d}_i のフーリエ変換である.

$$\hat{D}_i = \text{diag}(\hat{\mathbf{d}}_i) \quad (2.43)$$

$$\hat{\mathbf{D}} = (\hat{D}_1, \hat{D}_2, \dots, \hat{D}_M) \quad (2.44)$$

式 (2.42) は以下のように書き換えることができる.

$$\hat{\mathbf{x}}^{(j+1)} = (\hat{\mathbf{D}}^H \hat{\mathbf{D}} + \mathbf{I})^{-1} \left\{ \hat{\mathbf{D}}^H (\hat{\mathbf{g}}_0^{(j)} + \hat{\mathbf{f}} - \hat{\mathbf{h}}_0^{(j)}) + \hat{\mathbf{g}}_1^{(j)} - \hat{\mathbf{h}}_1^{(j)} \right\} \quad (2.45)$$

式 (2.45) の $\hat{\mathbf{D}}^H \hat{\mathbf{D}}$ はサイズが大きく計算量が大きくなってしまふ. そのため, 逆行列の補助定理を用いて計算する.

式 (2.38) と式 (2.39) はソフト閾値処理により求めることができる. 従って, 以下のように書き換えることができる.

$$\mathbf{g}_0^{(j+1)} = \text{sign}(\mathbf{a}) \odot \max(0, |\mathbf{a}| - \frac{1}{\rho}) \quad (2.46)$$

$$\mathbf{a} = \mathbf{D}\mathbf{x}^{(j+1)} - \mathbf{f} + \mathbf{h}_0^{(j)} \quad (2.47)$$

$$\mathbf{g}_1^{(j+1)} = \text{sign}(\mathbf{x}^{(j+1)} + \mathbf{h}_1^{(j)}) \odot \max(0, |\mathbf{x}^{(j+1)} + \mathbf{h}_1^{(j)}| - \frac{\lambda}{\rho}) \quad (2.48)$$

2.5 ノンキーフレームに対する係数行列を求める

$$\arg \min_{\mathbf{x}_m} \left\| \Phi \sum_{m=1}^M \mathbf{d}_m * \mathbf{x}_m - \mathbf{y} \right\|_2^2 + \mu \sum_{m=1}^M \|\mathbf{x}_m\|_1 \quad (2.49)$$

式 (2.33) と式 (2.34) を用いて式 (2.49) を以下のように書き換える.

$$\arg \min_{\mathbf{x}} \|\Phi \mathbf{D}\mathbf{x} - \mathbf{y}\|_1 + \lambda \|\mathbf{x}\|_1 \quad (2.50)$$

式 (2.50) は微分可能な関数と微分不可能な凸関数との和なので, 近接勾配法 [4] で解く.

近接勾配法とは微分可能な関数 f と微分不可能な凸関数 g の和の最小解を求めるアルゴリズムである.

$$\arg \min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) \quad (2.51)$$

更新式は以下のようになる.

$$\mathbf{x}^{(j+1)} = \text{prox}(\mathbf{z}) \quad (2.52)$$

$$\mathbf{z} = \mathbf{x}^{(j)} - \rho \nabla f(\mathbf{x}^{(j)}) \quad (2.53)$$

$$\text{prox} = \arg \min_{\mathbf{x}} g(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \quad (2.54)$$

式 (2.50) を解く際の $\nabla f(\mathbf{x})$ と prox は次式となる.

$$\nabla f(\mathbf{x}) = (\Phi \mathbf{D})^T (\Phi \mathbf{D}\mathbf{x} - \mathbf{y}) \quad (2.55)$$

$$\text{prox}(\mathbf{z}) = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \quad (2.56)$$

式 (2.56) はソフト閾値処理により求められる。従って、式は以下のように書き換えることができる。

$$\text{prox}(\mathbf{z}) = \text{sign}(\mathbf{z}) \odot \max(0, |\mathbf{z}| - \frac{\mu}{\rho}) \quad (2.57)$$

第3章

提案手法

提案手法では観測行列 Φ として，フーリエ変換行列 F に高周波成分を削除する行列 M_H を乗じたものを用いる．

$$\Phi = M_H F \quad (3.1)$$

式 (3.1) を式 (2.50) に代入すると以下となる．

$$\arg \min_x \left\| M_H \hat{D} \hat{x} - \mathbf{y} \right\|_2^2 + \mu \|\mathbf{x}\|_1 \quad (3.2)$$

辞書と係数の畳みこみ演算をフーリエ領域で計算できるようになるため，計算量を減らすことが可能となる．式 (3.2) を近接勾配法で解くと以下のようなになる．

$$\nabla f(\mathbf{x}) = (\hat{D} M_H)^H (M_H \hat{D} \hat{x} - \mathbf{y}) \quad (3.3)$$

$$\mathbf{z} = \mathbf{x}^{(j)} - \rho \nabla f(\mathbf{x}^{(j)}) \quad (3.4)$$

$$\mathbf{x}^{(j+1)} = \text{prox}(\mathbf{z}) = \text{sign}(\mathbf{z}) \odot \max(0, |\mathbf{z}| - \frac{\mu}{\rho}) \quad (3.5)$$

第4章

実験

動画 foreman (1 フレームあたり 128×128 画素) に対して実験を行った. 近接勾配法のループ回数は 10000 回とし, 辞書による精度の誤差を考慮するため, キーフレームから求めた辞書は既存手法と提案手法で同様のものを用いた. この条件下で圧縮率 MR を 0.5, 0.3 と変化させた際の, キーフレームと隣接する 200 枚のノンキーフレームの復元画像の SSIM 値と PSNR 値を調べる. パラメータ μ はランダムに選んだノンキーフレームを復号化した際に最も良い SSIM 値を出した値を使用している. それぞれの MR でのパラメータ μ を表 4.1 に示す. また, 既存手法と提案手法の実行時間とメモリ使用量を調べる.

表 4.1: 実験に使用したパラメータ μ の値

MR	従来法	提案法
0.5	0.000005	0.5
0.3	0.00001	1

第5章

実験結果・考察

実験結果を表 5.1, 図 5.1, 図 5.2, 図 5.3, 図 5.4 に示す. 原画像を図 5.5, 復元画像を図 5.6, 図 5.7 に示す. 表より, 従来手法に比べて提案手法が符号化, 復号化の際の実行速度, メモリ使用量を削減できていることが分かる. また, SSIM 値と PSNR 値においても従来手法よりも上回る精度を出していることが確認できる. 精度が良くなった理由として, 提案手法では高周波成分の情報を削減しているため, 従来手法に比べて削減する情報が少ないためであることが考えられる.

表 5.1: 符号と復号の際の実行速度とメモリ使用量

	従来法		提案法	
	符号	復号	符号	復号
実行速度 [秒]	0.0399	20.1	0.00108	4.83
メモリ使用量 [BiM]	1146	1163	122.9	150.5

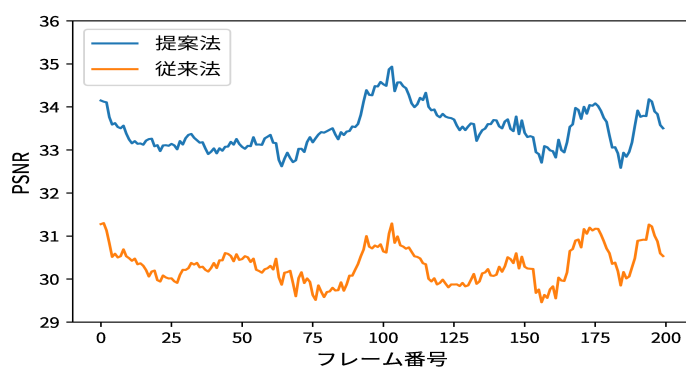


図 5.1: MR0.5 の際の PSNR 値

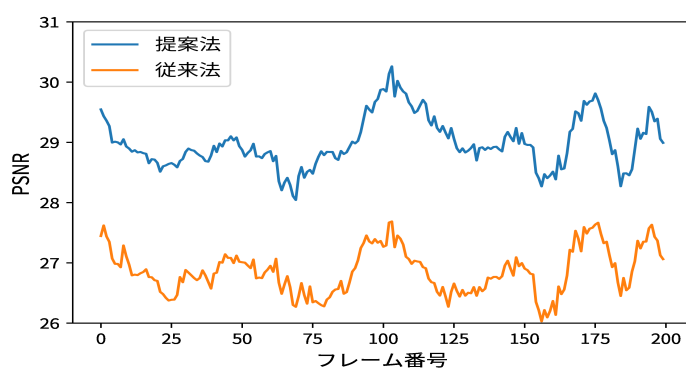


図 5.2: MR0.3 の際の PSNR 値

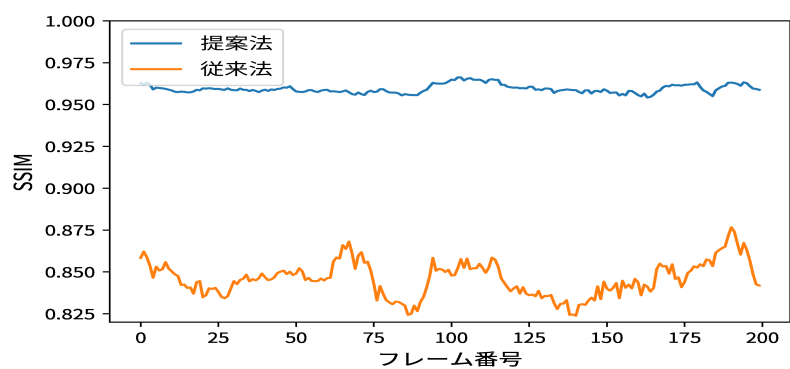


図 5.3: MR0.5 の際の SSIM 値

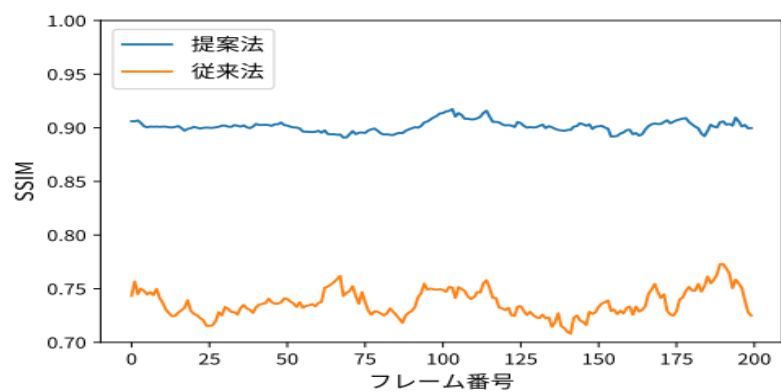


図 5.4: MR0.3 の際の SSIM 値



[1] キーフレーム

[2] ノンキーフレーム

図 5.5: 動画の時間軸上で隣接する原画像



[1] 従来法

[2] 提案法

図 5.6: MR0.5 での復元画像



[1] 従来法

[2] 提案法

図 5.7: MR0.3 での復元画像

第6章

むすび

本研究では，符号化と復号化の際の計算量とメモリ使用量削減のために観測行列にフーリエ行列を用いた CSC に基づく分散圧縮符号化を行った．実験の結果，符号化と復号化の際の計算量とメモリ使用量を削減できることが分かった．更に，提案手法で復号化することでより良い精度で復号化することが可能であると分かった．今後の課題として，係数推定の問題に L1 誤差を用いたものが挙げられる．

謝辞

本研究及びその基礎の学習において，貴重な時間を割いてご指導して頂いた久留米工業高等専門学校制御情報工学科の黒木祥光教授，および同研究室の専攻科生に深く感謝致します．

参考文献

- [1] C. Garcia-Cardona and B. Wohlberg, "Convolutional Dictionary Learning: A Comparative Review and New Algorithms," in IEEE Transactions on Computational Imaging, vol. 4, no. 3, pp. 366-381, Sept. 2018, doi: 10.1109/TCI.2018.2840334
- [2] 三村和史, "圧縮センシング－疎情報の再構成とそのアルゴリズム," 数理解析研究所講究録第 1803 巻, 京都, 2012
- [3] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," Computers Mathematics with Applications, vol.2, pp.17-40, Jan. 1976. DOI:10.1016/08981221(76)90003-1.
- [4] 永原 正章, "スパースモデリングのための凸最適化," システム／制御／情報, 61 巻, 1 号, pp. 20-28, Jul. 2017. https://doi.org/10.11509/isciesci.61.1_20