Vol.2017-CVIM-208 No.19
2017/9/15

# Deep Learningによる圧縮ビデオセンシングの再構成

吉田 道隆[1,a)]　長原 一[2]　鳥居 秋彦[3]　奥富 正敏[3]　谷口 倫一郎[1]

**概要**：スチルカメラは空間解像度の高い画像を撮影することができるが，フレームレートの高い撮影は行えない．一方で，ビデオカメラではフレームレートの高い時系列の画像を撮影できるが，空間解像度を高くすることは難しい．この様に，一般的なカメラの空間解像度とフレームレートにはトレードオフがあるが，これは撮像センサからのリードアウトやアナログデジタル変換などのハードウェアの制限に起因するものである．これを克服する方法として，圧縮ビデオセンシング手法が研究されている．圧縮ビデオセンシングでは，画素をランダムなタイミングで露光可能なセンサにより撮像したからランダム露光画像から，シーンのスパース性を利用して，高解像度で高フレームレートの動画を再構成する．この再構成処理にスパース最適化や Gaussian Mixture Models(GMM) を用いた手法など様々な手法が提案されてきたが，一般に再構成に時間がかかるという問題があった．それに対して，近年 Deep Learning による圧縮ビデオセンシングのための再構成手法が提案され高速化が実現出来ている．本研究では，この Deep Learning による再構成手法と従来再構成手法の比較実験を行い，その画質や復元時間についての評価を行った．また圧縮ビデオセンシングに用いる，ランダム露光パターンに関しても比較検討を行った．

# Reconstruction of compressed video sensing by Deep Learning

Michitaka YOSHIDA[1,a)]　Hajime NAGAHARA[2]　Akihiko TORII[3]　Masatoshi OKUTOMI[3]
Rin-ichiro TANIGUCHI[1]

## 1. はじめに

カメラでの動画の撮影には空間解像度とフレームレートにトレードオフがある．スチルカメラは空間解像度の高い画像を撮影することができるが，フレームレートの高い撮影は行えない．また，ビデオカメラではフレームレートの高い動画像を撮影できるが，空間解像度を高くすることは難しい．このトレードオフは，センサからのリードアウトやアナログデジタル (AD) 変換などのハードウェアの制限に起因するものである．そのため，バッファを設けて画素からのリードアウト時間を短縮することや，並列 AD 変換器を用いて AD 変換の時間を短縮するなど，特殊な構造のセンサを開発することでこのトレードオフを解消することができる．しかしながら，当然ながらこのような特殊なセンサは高価であることや，回路が複雑になることからフォトトランジスタの面積が減少するため感度が悪くなるといった問題もある．

一方で，一般的なセンサやカメラを組み合わせることでこのトレードオフを克服する研究もこれまでに提案されている．Wilburn ら [1] は多くの低フレームレートのカメラの露光タイミングを少しずつずらして撮影し，後処理を施すことで高フレームレートの動画を得ている．Gupta ら [2] はスチルカメラとビデオカメラを組み合わせて撮影し，ビデオカメラで撮影した各フレームをバイキュービック法でスチルカメラと同じ解像度まで拡大し，スチルカメラで撮影した最も近い 2 枚の画像と比較することで高空間解像度かつ高フレームレートの動画を得ている．しかし，これらの研究の観測システムは複数のカメラを用いるため装置が大きくなるほか，観測されるデータの冗長性が高いという問題がある．そこで，空間解像度とフレームレートのトレードオフを克服する手段の一つとして圧縮ビデオセンシング手法が提案されてきた [3], [4], [6], [7].

1　九州大学大学院システム情報科学研究院
2　大阪大学データビリティフロンティア機構
3　東京工業大学
a)　yoshida@limu.ait.kyushu-u.ac.jp

ⓒ 2017 Information Processing Society of Japan

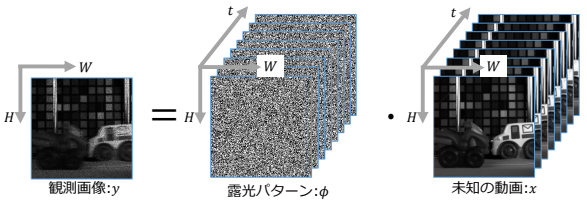



**図 1** 動画の圧縮センシングの観測モデル

通常の動画撮影は，すべての画素が同時に露光するグローバルシャッタの撮像センサを用いて静止画を連続撮影することで実現されるのに対して，圧縮ビデオセンシングは，画素の露光タイミングをランダムにずらした画像を撮影することで，より密な時間の情報をスパースに単一画像にサンプリングすることになる．このランダム露光画像から再構成処理により，撮像センサの時空間サンプリングを超えた画像を再構成する．この圧縮ビデオセンシングの観測過程は図 1 のようにモデル化できる．ここで，$\vec{x}$ は動的シーン，$\vec{y}$ はランダム露光による観測画像，$\vec{\phi}$ はランダム露光パターンである．これを行列で表すと以下の式のようになる．

$$\vec{y} = \vec{\phi}\vec{x} \tag{1}$$

圧縮ビデオセンシングでは，ランダム露光画像 $\vec{y}$ から未知の動画 $\vec{x}$ を再構成するが，観測された画像 $\vec{y}$ の要素数は $W \times H$ であり，未知の動画 $\vec{x}$ は要素数 $W \times H \times t$ であるため，一意に求めることはできない．Hitomi ら [6] や Sonoda ら [3] は適切な動画を学習して得た辞書による基底とその係数で再構成したい動画を表現し，係数をスパース最適化によって求めることで動画を再構成している．しかし，スパース最適化は NP 困難な問題であることが知られており，このような再構成は一般的に膨大な計算時間がかかり実用的ではない．Yang ら [7] は，動画が Gaussian Mixture Models(GMM) で表せると仮定し，観測画像を条件とする条件付き期待値を求めることで動画のより高速な再構成を実現した．また，Iliadis ら [4] は Deep Learning を用いたデコーダを提案し，さらなる再構成の高速化を実現した．これらのように，これまでに様々な圧縮ビデオセンシングの再構成手法が提案されてきた．また，圧縮ビデオセンシングに必要な露光パターンについても，ブロックランダムパターンやハードウェア実装を考慮した疑似ランダムパターンが提案されてきた [3], [4], [6]．本研究では，これら圧縮ビデオセンシング手法での動画の再構成時間や復元品質についての比較検討する．また，3 種類のランダム露光パターンを比較することで，同様にその実現性と復元画質について検討する．

## 2. 圧縮ビデオセンシングのためのランダム露光パターン

式 1 に示したように，圧縮ビデオセンシングの入力としてランダム露光された観測画像が必要となる．しかしながら，一般的な CCD や CMOS センサは，すべての画素が同時露光するグローバルシャッタや画素の読み出し順に露光を行うローリングシャッタの撮像センサが一般的であり，ランダム露光可能な市販センサは市場には存在しない．そのため，これまでの圧縮ビデオセンシングや再構成の研究では，理想的なランダム露光を想定したものや，ハードウェア実装上の制限を考慮した疑似ランダムパターンが露光パターンとして用いられてきた．

Iliadis ら [4] は，各画素の露光時間を 16 分割し，それを 8/16 でランダム配置したパターンを $4 \times 4$ のブロックパターンとした時間ランダム露光パターンを提案した．この $4 \times 4$ のパターンを繰り返した $8 \times 8$ のパターンを 1 パッチとして $8 \times 8$ のランダム露光画像から $8 \times 8 \times 16$ の動画をパッチ毎に Deep learning を用いたデコーダで復元することで，動画全体を再構成した．この論文では，このような露光が可能な理想的なセンサを想定しているため，シミュレーション実験による評価にとどまっている．

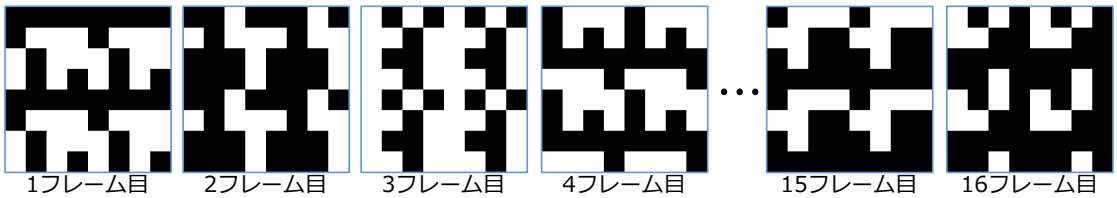
Hitomi らは [6]，一般的な非破壊読み出しができない CMOS センサの構造を想定して，同一の露時間で 1 フレームにおいて一回の露光ではあるが，露光開始および終了タイミングをランダムとし，それを $7 \times 7$ パッチとしたランダム露光パターンを提案した．このパターンは，CMOS 実装上の制限を想定したものであるが，論文では残念ながらシミュレーションおよび反射光学系と Liquid Crystal on Silicon(LCoS) を用いた疑似実装による実験しか行われていない．

Sonoda らはピクセルごとに露光を制御できるプロトタイプの CMOS センサを用いて露光パターンを実装した．このセンサは，このプロトタイプ CMOS センサは 8 本の Reset 信号線と 8 本の Transfer 信号線を持ち，それぞれ $8 \times 8$ のパッチ毎に縦列および横列で制御を行うことができる．この CMOS センサを用いた疑似ランダム露光を提案した．このパターンでは，完全ランダム露光に対して，性能は若干劣るものの，実際にこのプロトタイプ CMOS を用いた圧縮ビデオセンシングを実現している．

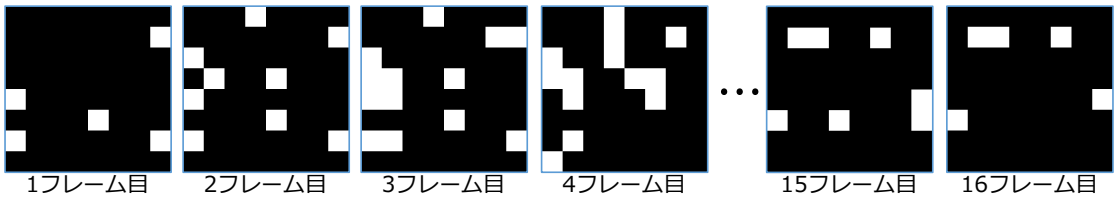
本研究では，これら 3 種類の露光パターンについて比較検討を行う．公平な評価を行うために，上記の生成アルゴリズムやハードウェア実装上の制限を踏襲した上で，$8 \times 8 \times 16$ のパッチ露光パターンとした．図 2 にそれぞれの手法の露光パターンを例として示した．これらの $8 \times 8 \times 16$ のパッチの繰り返しパターンを全画像の露光パターンとしてランダム撮像画像の生成に用いた．

## 3. 圧縮ビデオセンシングの再構成アルゴリズム

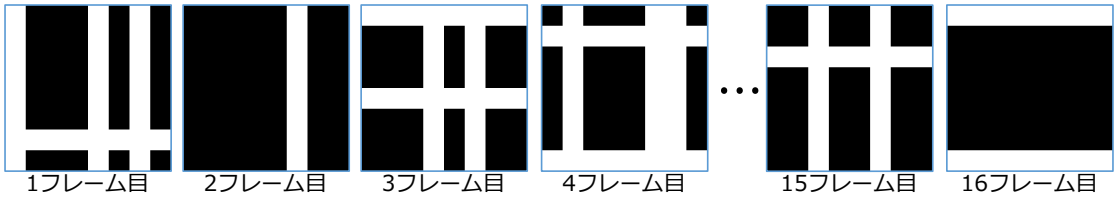
1 章および式 1 で示した様に，ランダム露光観測画像は，$W \times H$ の画像として得られる．しかしながら，画像全体




(a)Iliadis らの露光パターン




(b)Hitomi らの露光パターン




(c)Sonoda らの露光パターン

**図 2** 各露光パターンの例

を一度に再構成するのは，計算コストの観点からも現実的ではないため，一般的には $W_p \times H_p$ のパッチ毎に計算される．ここでは，本論文で比較する $W_p \times H_p$ にパッチ分割された観測画像パッチ $y_p$ から $W_p \times H_p \times T$ の復元パッチボリューム $x_p$ を復元する手法について説明する．

### 3.1 スパース最適化による再構成

スパース最適化による再構成では，復元する動画パッチボリュームが式 2 に示す様に，スパースな基底で表現できると仮定する．

$$\vec{x_p} = \vec{D}\vec{\alpha} = \alpha_1 D_1 + \alpha_2 D_2 + \cdots + \alpha_k D_k \qquad (2)$$

$\vec{\alpha} = [\alpha_1, .., \alpha_k]^{\mathrm{T}}$ は係数であり，係数の個数 $k$ は観測画像より低次元であると仮定する．Hitomi ら [6] の手法では，$D_1, \ldots, D_k$ は適切な動画から KSVD 法によって得られた基底辞書 $\vec{D}$ を用いている．式 (1)，(2) より，$\vec{y_p} = \vec{\phi_p}\vec{D}\vec{\alpha}$ が得られる．$\vec{y_p}$，$\vec{\phi_p}$，$\vec{D}$ は既知のため，$\vec{\alpha}$ を求める問題となる．ここで $\vec{\alpha}$ はスパースであると仮定すると，スパース最適化により $\vec{\alpha}$ を求めることができる．

Hitomi ら [6] はスパース最適化手法として Orthogonal Matching Pursuit(OMP) アルゴリズム [10] を用いた．OMP アルゴリズムは $L_0$ ノルム正則化を行うアルゴリズムで，以下の式を解くアルゴリズムである．

$$\vec{\alpha} = \arg\min_{\vec{\alpha}} ||\vec{\alpha}||_0 \text{ subject to } ||\vec{\phi}\vec{D}\vec{\alpha} - \vec{y_p}||_2 \leq \sigma \qquad (3)$$

ここで $\sigma$ は許容誤差である．OMP アルゴリズムではあらかじめ定めた $M$ 個の再構成に用いる基底とその係数を求める．再構成に使う基底の選び方はまず，まだ使われていない基底の中から残差との内積が最も大きくなるものを一つ選び，再構成に使う基底群に加える．誤差が小さくなるよう係数を決め，新たな残差を求める．この操作を残差が一定以下となるまで繰り返すか，$M$ 回繰り返すことで係数 $\vec{\alpha}$ を求め，動画 $\vec{x_p} = \vec{D}\vec{\alpha}$ を再構成することができる．

また，本研究ではスパース最適化として SPGL1[14] も用いる．SPGL1 は以下の式 (4) を解くことで係数ベクトルの $L_1$ ノルム正則化を行い，係数ベクトルを決定することで動画を再構成する．

$$\vec{\alpha} = \arg\min_{\vec{\alpha}} ||\vec{\phi}\vec{D}\vec{\alpha} - \vec{y_p}||_2 \text{ subject to } ||\vec{\alpha}||_1 \leq \tau \qquad (4)$$

ここで $\tau$ は係数の合計の上限である．SPGL1 は $||\vec{\phi}\vec{D}\vec{\alpha} - \vec{y_p}||_2$ と $||\vec{\alpha}||_1$ をグラフにプロットし，ニュートン法を応用した手法により，$||\vec{\alpha}||_1 \leq \tau$ を満たしながら誤差を最小化する $\vec{\alpha}$ を求める．

### 3.2 GMM による再構成

Yang ら [7] は GMM モデルを用いた圧縮ビデオセンシングのための再構成手法を提案している．ここでは，式 (1) の $\vec{y}$ をパッチに分け，ノイズを加えた

$$\vec{y_p} = \vec{\phi_p}\vec{x} + \epsilon_p \qquad (5)$$

について考える．ここで，$\epsilon_p \in \mathbb{R}^{W_p \times H_p}$ はノイズである．まず，パッチに分けた動画が $\{\vec{x}_p\}$ が以下のように Gaussian Mixture Models(GMM) で表せると仮定する．

$$\vec{x}_p \sim \sum_{k=1}^{K} \lambda_k \mathcal{N}(\vec{x}_p \mid \vec{\mu}_k, \vec{\Sigma}_k) \tag{6}$$

ここで，$\mathcal{N}$ はガウス分布であり，$K$，$\vec{\mu}_k$，$\vec{\Sigma}_k$，及び $\lambda_k$ はそれぞれ，GMM の要素数，平均，共分散行列，及び $k$ 番目の混合係数である ($\lambda_k > 0$ かつ $\sum_{k=1}^{K} \lambda_k = 1$)．Yang らは動画を表す GMM のパラメータ ($\vec{\mu}_k$，$\vec{\Sigma}_k$，$\lambda_k$) を訓練データから EM アルゴリズムにより求ている．

観測モデルを $\epsilon_p \sim \mathcal{N}(0, \vec{R})$ ($\vec{R}$ は $(W_p \times H_p) \times (W_p \times H_p)$ の共分散行列) とすると観測された画像が

$$\vec{y}_p \mid \vec{x}_p \sim \mathcal{N}(\vec{y}_p \mid \vec{\phi}_p \vec{x}_p, \vec{R}) \tag{7}$$

と表されるため，ベイズの定理を適用すると以下の式のようになり，これも GMM となる．

$$p(\vec{x}_p \mid \vec{y}_p) = \sum_{k=1}^{K} \tilde{\lambda}_k \mathcal{N}(\vec{x}_p \mid \tilde{\vec{\mu}}_k, \tilde{\vec{\Sigma}}_k) \tag{8}$$

ここで，

$$\tilde{\lambda}_k = \frac{\lambda_k \mathcal{N}(\vec{y}_p \mid \vec{\phi}_p \vec{\mu}_k, \vec{R} + \vec{\phi}_p \vec{\Sigma}_k \vec{\phi}_p^T)}{\sum_{l=1}^{K} \lambda_l \mathcal{N}(\vec{y}_p \mid \vec{\phi}_p \vec{\mu}_l, \vec{R} + \vec{\phi}_p \vec{\Sigma}_l \vec{\phi}_p^T)} \tag{9}$$

$$\tilde{\vec{\Sigma}}_k = (\vec{\phi}_p^T \vec{R}^{-1} \vec{\phi}_p + \vec{\Sigma}_k^{-1})^{-1} \tag{10}$$

$$\tilde{\vec{\mu}}_k = \tilde{\vec{\Sigma}}_k (\vec{\phi}_p^T \vec{R}^{-1} \vec{y}_p + \vec{\Sigma}_k^{-1} \vec{\mu}_k) \tag{11}$$

である．よって $\vec{x}_p$ の条件付期待値を求めることで動画を復元することができる．

$$\mathbb{E}(\vec{x}_p \mid \vec{y}_p) = \int \vec{x}_p p(\vec{x}_p \mid \vec{y}_p) d\vec{x}_p = \sum_{k=1}^{K} \tilde{\lambda}_k \tilde{\vec{\mu}}_k \stackrel{\text{def}}{=} \tilde{\vec{x}}_p \tag{12}$$

### 3.3 Deep Learning による再構成

本研究の Deep Learning による手法は Iliadis ら [4] の研究に基づいている．Deep Learning を用いた手法は Deep Learning によって動画を再構成するデコーダの学習を行う事前処理と，学習したデコーダを用いて圧縮センシングにより観測された画像から動画を再構成する処理に分けられる．Deep Learning とは多層のニューラルネットワークを用いた機械学習で，非常に多くの学習データを必要とするが特徴量を自動で計算し学習を行うものである．時間軸方向に圧縮して観測された画像から動画を再構成するネットワークを学習する．ネットワークの学習により各層をつなぐノードの重みとバイアスが更新され，特徴量が学習される．デーコーダの学習はパッチごとに行う．訓練データの動画をパッチに分けたそれぞれを $\vec{x_p}, p \in \mathbb{N}$ とすると，ランダム露光観測された画像のパッチは $\vec{y_p} = \vec{\phi_p} \vec{x_p}$ と表せる．それぞれの大きさは $\vec{x_p} : W_p \times H_p \times T$，$\vec{y_p} : W_p \times H_p$，$\vec{\phi_p} : W_p \times H_p \times T$ であり，$\vec{\phi_p}$ は全パッチで共通である．

Iliadis らは観測した画像 $\vec{y_p}$ から動画 $\vec{x_P}$ を再構成するためのネットワークに Multi-Layer perceptron(MLP) を用いている．この MLP の構造は入力層の大きさが $W_p \times H_p$，出力層の大きさが $W_p \times H_p \times T$ であり，隠れ層 $L_k, k = 1, \ldots, 4$ も同様の $W_p \times H_p \times T$ である 4 層の全結合層である．動きを学習するには 3 次元のたたみ込み層が有効である [11]．初めの隠れ層は 2 次元の入力と 3 次元の全結合層であり，残り 3 層は本質的に 3 次元である出力層と同じ大きさであるため，3 次元のたたみ込み層とみなすことができる．ネットワークの各隠れ層は以下の式で定義されている．

$$h_k(\vec{y}) = \sigma(\vec{b_k} + \vec{\omega_k} \vec{y}) \tag{13}$$

ここで，$\vec{b_k}$ はバイアスベクトルであり，$\vec{\omega_k}$ は出力の重み行列である．また，非線形関数 $\sigma()$ は以下で定義される Rectified Linear Unit(ReLU) である．

$$\sigma(y) = max(0, y) \tag{14}$$

パラメータ $\vec{\theta} = \{\vec{b_{0,\ldots,K}}, \vec{\omega_{0,\ldots,K}}\}$ は観測画像をネットワークに通した出力 $f(\vec{y_i}; \theta)$ と $\vec{x_i}$ の二乗誤差が小さくなるようにバックプロパゲーションにより更新する．損失関数には以下の平均二乗誤差 (MSE) を用いる．

$$L(\vec{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \| f(\vec{y_i}; \theta) - \vec{x_i} \|_2^2 \tag{15}$$

損失関数に MSE を用いるのは再構成結果の評価に MSE と関係の深いピーク信号対雑音比 (PSNR) を活用するためである．

動画再構成は，ランダム露光観測画像をパッチごとに再構成のためのデコーダに入力し，学習した重みを使って結果を出力することで再構成動画を得る．この Deep Learning による手法は観測画像をパッチに切り分け学習された重みとのたたみ込みを計算するだけであり，非常に高速な処理が可能である．

## 4. 実験

### 4.1 実験設定

訓練データはすべての手法で同様のものを用いた [15]．これは，映像要約のベンチマーク用のデータセットで，この中の 20 本の動画から 16 フレームをランダムに 4 シーンずつ取り出し，それぞれに回転 $(90°, 180°, 270°)$ と反転を行ったものを用いた．2 章で示した様に入力とする露光パターンは，8 の繰り返しパッチであり，入力とするランダム観測画像 $W \times H$ は，この $8 \times 8$ の繰り返しとして露光されている．現実的な計算時間での復元処理のため，再構成のためのパッチサイズを $W_p = H_p = 8$, $T = 16$ とし，いずれの手法も $8 \times 8$ のパッチ毎に復元処理を行い，これ

らを結合することで全画像を復元するものとする．再構成時間及び再構成品質の比較は，空間解像度 $256 \times 256$ の 16 フレームの 14 本の動画をそれぞれの露光パターンで 1/16 に圧縮し，それぞれの手法で空間解像度 $256 \times 256$ の 16 フレームの動画に再構成することで行った．使用した動画は人物行動推定用データセットである UCF-101[16] や，Sonoda ら [3] がハイスピードカメラで撮影した動画なのである．再構成品質は PSNR によって評価した．露光パターンは Iliadis ら [5] のもの，Hitomi ら [6] のものおよび Sonoda ら [3] のものを用いた．各露光パターンを図 2(a), (b), (c) に示す．本研究では，これらのパターンのランダム露光センサをシミュレートして撮影画像を作成した．

比較する手法は以下の通りある．

- OMP[10]：動画を学習により得た基底とその係数で表す．係数ベクトルはスパースであるため，式 (3) を解くことで係数ベクトルの $L_0$ ノルム正則化を行い，係数ベクトルを決定することで動画を再構成する．式 (3) における許容誤差 $\sigma$ は事前実験の結果より，0.01 とした．学習した基底の数は事前実験で最も性能の良かった 5K とした．
- SPGL1[14]：OMP と同様に動画を学習により得た基底とその係数で表す．係数ベクトルはスパースであるため，以下の式 (4) を解くことで係数ベクトルの $L_1$ ノルム正則化を行い，係数ベクトルを決定することで動画を再構成する．式 (4) における係数の合計の上限 $\tau$ は事前実験の結果より 64 とした．動画の基底は OMP と同じ 5K のものを用いた．
- GMM：パッチに分けた動画が GMM により表現できると仮定し，訓練データから動画を表す GMM を EM アルゴリズムにより学習する．観測画像にベイズの定理を適用するとこれも GMM で表せるので，条件付き期待値を求めることで動画を再構成する．GMM の要素数は事前実験で最も性能の良かった 40 とした．
- Deep Learning: 動画を再構成するデコーダを Deep Learning により学習し，観測した画像をデコーダに通すことで動画を再構成する．本研究で用いた MLP の構造は入力層の大きさが $8 \times 8 = 64$，出力層の大きさが $8 \times 8 \times 16 = 1024$ であり，隠れ層 $L_k, k = 1, \ldots, 4$ が大きさ 1024 である 4 層の全結合層である．つまり，このデコーダは時間軸方向に圧縮された $8 \times 8$ の空間解像度の画像を受け取り，16 フレームを再構成して $8 \times 8 \times 16$ の時系列の画像を出力する．

OMP，SPGL1 で再構成を行う際には $8 \times 8 \times 16$ のパッチを縦横 1 ピクセルずつずらして再構成を行い，それぞれの平均を再構成結果とする．同様に GMM で再構成を行う際にはパッチを $4 \times 4$ ピクセルずつずらし再構成を行い，それぞれの平均を再構成結果とする．Deep Learning では，露光パターンがネットワークに学習されてしまい，パッチに対し露光パターンが固定されてしまう．パッチをずらして再構成を行う際には露光パターンもずらす必要があるため，Deep Learning による手法ではパッチをずらすときには別のネットワークを学習する必要がある．よって Deep Learning よる再構成ではそのような処理はせず，パッチを並べることで再構成結果とする．

### 4.2 再構成手法の比較

14 本の動画に対し再構成を行った結果の平均 PSNR とその分散，および平均再構成時間を表 1 に示す．また，図 3，4，5 に 3 つのシーンにおける，それぞれの露光パターンのランダム露光センサをシミュレートして作成したランダム露光画像から，各再構成手法により再構成した 16 フレームの動画の 1 フレーム目を示す．

表 1 より，Iliadis らの露光パターンおよび Sonoda らの露光パターンにおいては GMM を用いた手法が最も PSNR が高く，次いで Deep Learning による手法が高かった．また，Hitomi らの露光パターンにおいては Deep Learning による手法が最も PSNR が高く，次いで GMM を用いた手法が高かった．どの露光パターンにおいても GMM を用いた手法と Deep Learning を用いた手法の PSNR の差は 1dB 以下であるが，これら 2 手法とスパース最適化を用いた 2 手法との差は 2dB 以上ある．また，スパース最適化による 2 手法はほかの 2 手法と比べ，再構成に非常に時間がかかるという結果となった．どの露光パターンにおいても再構成時間は Deep Learning による手法が最も短かった．

図 3，4，5 の 3 段目を見ると，OMP を用いた手法では再構成結果が全体的に細かなノイズが乗ったような画像となっており，画質があまりよくない．これは，係数の $L_0$ ノルム正則化を露光パターンで圧縮された画像に対して行うため，オーバーフィッティングが起こっていることが原因と考えられる．

図 3，4，5 の 4 段目を見ると，SPGL1 を用いた手法による再構成結果はモーションブラーが起こっているような画像となっている．これは，SPGL1 が $L_0$ ノルム正則化を行うため，係数ベクトルが完全にスパースとならずに様々な基底の情報が含まれているためと考えられる．

図 3，4，5 の 5 段目を見ると，GMM を用いた手法では，動いている部分はよく再構成できているが，静止している部分はちらつきが見らる．静止部分については GMM を用いた手法より最下段の Deep Learning を用いた手法のほうがよく再構成できている．

図 3，4，5 の最下段を見ると，Deep Learning による手法ではパッチの境界部分が連続的に変化していないため，ブロック状の模様が出ている．これは，露光パターンがネットワークに学習されてしまい，パッチに対し露光パターンが固定されてしまうため，一つのデコーダでは再構成の際にパッチをずらし平均をとるということができないためで

ある．また，Deep Learning による手法は他の手法と比べ再構成時間は短いものの，学習に時間がかかるという問題もある．しかし，観測しながら再構成を行うような場面ではほかの手法では再構成が間に合わないため，用途によって再構成手法を使い分ける必要がある．

表 1 より，GMM を用いた手法ではほかの手法に比べ分散が大きい．つまり，GMM を用いた手法は再構成品質がシーンに依存すると考えられる．

### 4.3 露光パターンの比較

スパース最適化の 2 手法および Deep Learning を用いた手法では Hitomi らの露光パターンが最も PSNR が高く，次いで Iliadis らの露光パターン，そして Sonoda らの露光パターンが最も PSNR が低かった．また，GMM を用いた手法では Iliadis らの露光パターンが最も PSNR が高く，次いで高かったのは Sonoda らの露光パターンであり，最も PSNR が低かったのは Hitomi らの露光パターンという結果となった．

図 3，4，5 の 1 列目を見ると，Iliadis らの露光パターンによりランダム露光された画像から再構成した動画は，動きのある部分にモーションブラーが起こっているような画像となっている．また，ブラーの部分は露光パターンの各ピクセルの露光回数の差によるアーティファクトが現れている．Iliadis らは再構成をパッチサイズ $8 \times 8 \times 16$ で行っているものの，露光パターンの大きさは $4 \times 4 \times 16$ となっている．この露光パターンはピクセルごとに露光を制御できる理想的なセンサを想定しており，ピクセルごとの制約が最も少ないため保存できる時間的な情報は多い．しかし，露光パターンの大きさがほかの露光パターンと比べ小さいため，保存できる空間的な情報は少ないと考えられる．

図 3，4，5 の 2 列目を見ると，Sonoda らの露光パターンによりランダム露光された画像から再構成した動画は，全体的にぼやけたような画像となっており，空間解像度があまり高くない．Sonoda らの露光パターンは露光を縦または横の列単位で制御できるセンサへ実装することを前提に設計されており，露光パターンにも縦および横の列に依存性があるため，ピクセル単位での露光の制御はできない．よって，露光パターンのランダム性に限界があるため，再構成品質があまりよくないと考えられる．

図 3，4，5 の 3 列目を見ると，Hitomi らの露光パターンによりランダム露光された画像から再構成した動画は，Iliadis らの露光パターンと同様に動きのある部分にモーションブラーが起こっているような画像となっている．しかし，各ピクセルの露光回数が同じため，Iliadis らの露光パターンと比べ，アーティファクトは抑えられている．Hitomi らは，ピクセルごとに露光を制御できるが，非破壊読み出しできないセンサを用いることを前提としているため，空間的なランダム性は本研究で比較した 3 つの露光パターンの中で最も高いものの，一つの連続した区間でしか露光できないという制約がある．これにより，一つのピクセルが記録できる時間的な情報が少なくなるが空間的な情報は多く保存できるため，再構成品質が比較的良いと考えられる．

これらの露光パターンの制約は実装方法に強く依存するため，実機に実装する際には露光パターンの自由度によって再構成品質がある程度決まってしまう．Hitomi らのように反射光学系と LCoS を用いるとピクセルごとに露光を制御できるが，装置の大型化や光学系の調整がシビアであるなどの問題もある．一方，Sonoda らの用いているセンサは露光を列単位でしか制御できないが，センサのみで露光パターンを実装でき，撮影は比較的簡単に行える．このように，露光パターンの性能と撮影装置の間にもトレードオフが存在する．近年では最適な露光パターンを Deep Learning を用いて設計する手法が提案されている [5]．Deep Learning により露光パターンとデコーダを同時に学習することにより，最適な露光パターンを見つけることができる．しかし，この手法では露光パターンが訓練データに最適化されてしまう，制約の強い露光パターンはネットワークの設計が困難といった問題がある．

## 5. まとめ

本研究では，近年圧縮センシングにも取り入れられつつあるディープラーニングを用いた手法も含め，様々な圧縮センシングによって撮影された画像から動画を再構成する手法の比較を行った．スパース最適化による手法として $L_0$ ノルム正則化を行う OMP と $L_1$ ノルム正則化を行う SPGL1，動画が GMM によって表現できると仮定し再構成をこなう手法，および Deep Learning によりデコーダを学習して再構成を行う手法について比較を行い，再構成品質は GMM を用いた手法と Deep Learning による手法がよく，再構成時間は Deep Learning による手法が最も短いという結果となった．

Deep Learning による手法において現在のネットワーク構造ではパッチの周りの情報を使うことができないため，パッチの境界が不連続になりブロックノイズが出てしまう．よって，ネットワークの入力に注目パッチの周りの画素も含めるようにするなど，ネットワークの構造をパッチの周りの情報を使うように変更することで，このようなブロックノイズを低減できると考えられる．

露光パターンについても実装方法により様々なものが提案されている．これは，露光パターンの設計は実装方法に大きく依存し，露光パターンの性能と撮影装置の間にはトレードオフが存在するためである．本研究では，Iliadis ら [4] の露光パターン，Hitomi ら [6] の露光パターン，Sonoda ら [3] の露光パターンを比較し，Hitomi らの露光パターンが最も性能が良いという結果となった．

表 1　各手法の再構成品質 (PSNR) とその分散，および再構成時間の平均

| | Iliadis | | | Sonoda | | | Hitomi | | |
|---|---|---|---|---|---|---|---|---|---|
| 手法 | PSNR | PSNR の分散 | 再構成時間 | PSNR | PSNR の分散 | 再構成時間 | PSNR | PSNR の分散 | 再構成時間 |
| OMP | 23.2706 | 12.1679 | 749.22 秒 | 22.8621 | 11.4284 | 743.42 秒 | 24.5664 | 11.2446 | 579.08 秒 |
| SPGL1 | 24.2845 | 13.1409 | 10492 秒 | 23.7131 | 13.1264 | 10152 秒 | 25.0052 | 12.9807 | 3691.6 秒 |
| GMM | 28.6337 | 15.9424 | 161.71 秒 | 28.1769 | 15.1816 | 157.76 秒 | 27.6914 | 13.8926 | 150.92 秒 |
| Deep Learning | 28.2352 | 13.8026 | 0.5658 秒 | 27.8053 | 13.3678 | 0.5764 秒 | 28.6486 | 13.6015 | 0.7570 秒 |

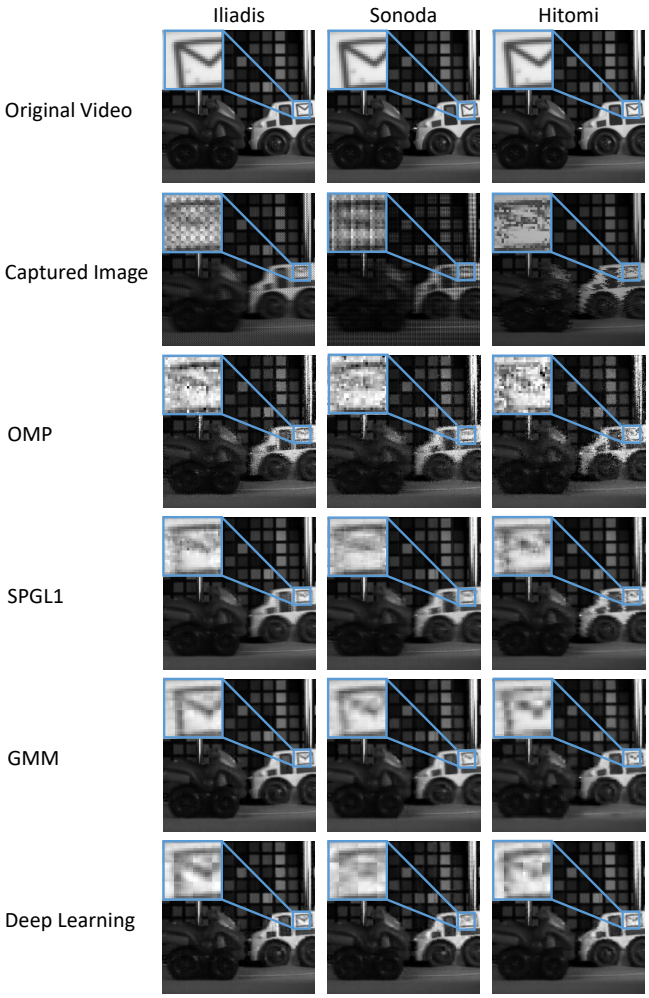



図 3　再構成結果 (Car)

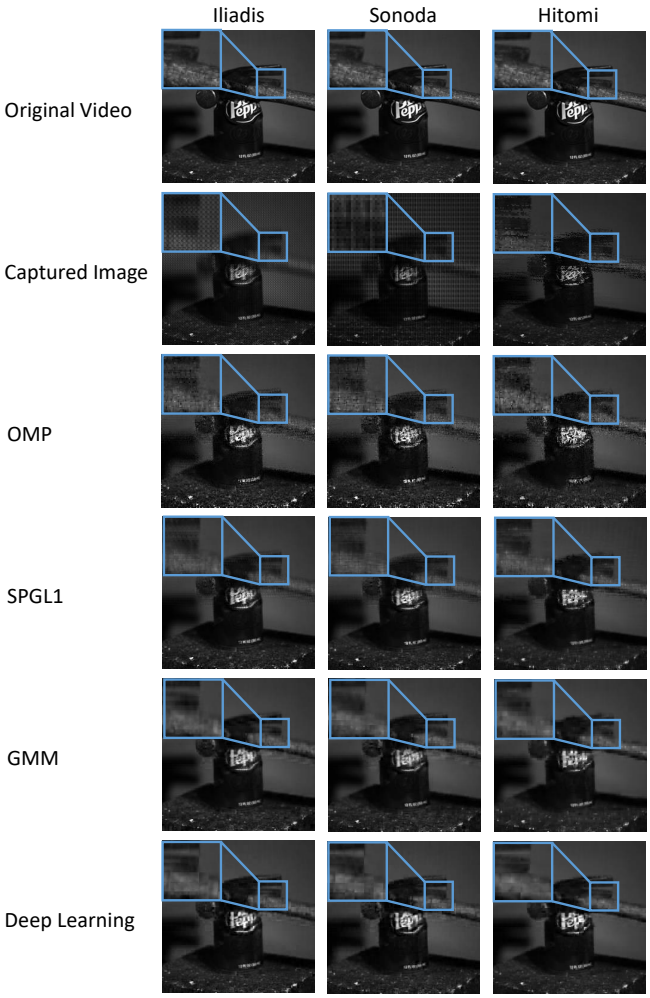



図 4　再構成結果 (Hammer)

ⓒ 2017 Information Processing Society of Japan

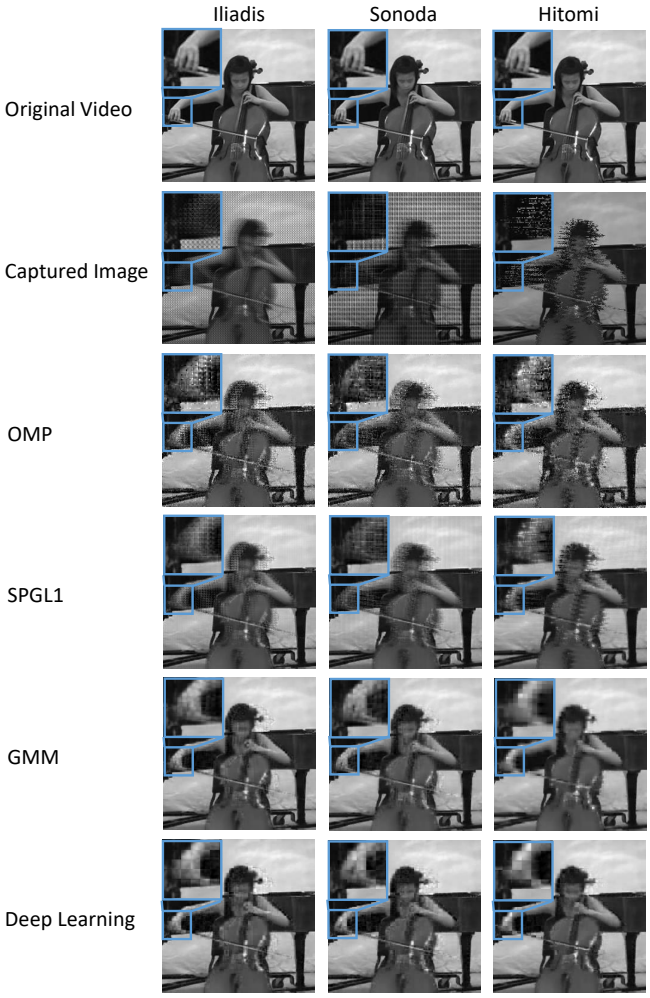



図 5　再構成結果 (Cello)

## 参考文献

[1] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, M. Horowitz, "High-speed videography using a dense camera array", Computer Vision and Pattern Recognition (CVPR),pp.II–294,2004.

[2] A. Gupta, P. Bhat, M. Dontcheva, O. Deussen, B. Curless, M. Cohen, "Enhancing and experiencing spacetime resolution with videos and stills", International Conference on Computational Photography (ICCP),pp.1–9,2009.

[3] T. Sonoda, H. Nagahara, K. Endo, Y. Sugiyama, R. Taniguchi, "High-speed imaging using CMOS image sensor with quasi pixel-wise exposure", International Conference on Computational Photography (ICCP),pp.1–11,2016.

[4] M. Iliadis, L. Spinoulas, A. K. Katsaggelos, "Deep Fully-Connected Networks for Video Compressive Sensing", arXiv preprint arXiv:1603.04930,2016.

[5] M. Iliadis, L. Spinoulas, A. K. Katsaggelos, "DeepBinaryMask: Learning a Binary Mask for Video Compressive Sensing." , arXiv preprint arXiv:1607.03343 (2016).

[6] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary", International Conference on Computer Vision (ICCV),pp.287–294,2011.

[7] J. Yang, X. Yuan, X. Liao, P. Llull, D. J. Brady, G. Sapiro, L. Carin, "Video compressive sensing using Gaussian mixture models", IEEE Transactions on Image Processing,pp.4863–4878,2014.

[8] J. Holloway, A. C. Sankaranarayanan, A. Veeraraghavan, S. Tambe, "Flutter shutter video camera for compressive sensing of videos", International Conference on Computational Photography (ICCP),pp.1–9,2012.

[9] J. Gu, Y. Hitomi, T. Mitsunaga, S. Nayar, "Coded rolling shutter photography: Flexible space-time sampling", International Conference on Computational Photography (ICCP),pp.1–8,2010.

[10] Y. C. Pati, R. Rezaiifar, P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers,pp.40–44,1993.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, "Learning spatiotemporal features with 3d convolutional networks", International Conference on Computer Vision (ICCV),pp.4489–4497,2015.

[12] J. Holloway, A. C. Jason, A. Veeraraghavan, "Flutter shutter video camera for compressive sensing of videos." , Computational Photography (ICCP), 2012 IEEE International Conference on. IEEE, 2012.

[13] R. Raskar, A. Agrawal, J. Tumblin, "Coded exposure photography: motion deblurring using fluttered shutter.", ACM Transactions on Graphics (TOG). Vol. 25. No. 3. ACM, 2006.

[14] E. van den Berg, M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions." , SIAM Journal on Scientific Computing 31.2 (2008): 890-912.

[15] M. Gygli, H. Grabner, H. Riemenschneider, L. V. Gool , "Creating Summaries from User Videos", ECCV, 2014, https://people.ee.ethz.ch/ gyglim/vsum/

[16] Y.-G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, R. Sukthankar, THUMOS Challenge: Action Recognition with a Large Number of Classes, 2013, http://crcv.ucf.edu/ICCV13-Action-Workshop

ⓒ 2017 Information Processing Society of Japan