

JPEG XLの決定木学習を利用した画像集合の可逆圧縮手法の検討

櫻井 拓也[†] 井上 潮^{††}

[†] 東京電機大学 工学研究科 情報通信工学専攻 〒120-8551 東京都足立区千住旭町 5 番

^{††} 東京電機大学 工学部 情報通信工学科 〒120-8551 東京都足立区千住旭町 5 番

E-mail: [†]20kmc08@ms.dendai.ac.jp, ^{††}inoueu@mail.dendai.ac.jp

あらまし インターネット上には多くの画像データセットが公開されており、コンピュータビジョンの発展に貢献している。巨大なデータセットは研究に役立つ反面、データ量が多いため転送に時間がかかるという問題がある。可逆圧縮はデータを損失せずに圧縮できる手法なため、広く応用が期待できる。しかし、可逆圧縮は非可逆圧縮よりも圧縮率が悪い。そこで、データセットの圧縮において、データセットの特徴を生かして複数の画像をまとめて圧縮することで、圧縮率が改善できると考えられる。本稿では、最新の画像圧縮手法である JPEG XL をベースに、その特徴的な符号化手法を利用した画像集合の可逆圧縮手法を提案する。大量の画像を現実的な計算量で圧縮するため事前にクラスタリングを行い、クラスタ内で画像どうしの最適な圧縮方法を探索する。また、従来の画像集合圧縮で用いられてきた手法と比較を行う。

キーワード 画像圧縮, 可逆圧縮, JPEG XL, 最小全域木

1 はじめに

インターネット上には多くの画像データセットが公開されており、コンピュータビジョンの発展に貢献している。訓練データの量が多いほど、コンピュータビジョンタスクの性能が向上することが報告されており [1], 巨大な画像データセットはコンピュータビジョンの発展に重要である。一方で、巨大な画像データセットは、データ量が多いため転送に時間がかかるという問題がある。例えば、ImageNet [2] は、一般公開されているサブセットだけでも、約 60 万枚の画像があり、その大きさは 166 GB にも及ぶ。したがって、このようなデータセットを記憶したり転送したりするには、データを圧縮することが必要である。

画像を圧縮する手法は、非可逆圧縮 (*lossy compression*) と可逆圧縮 (*lossless compression*) に分類できる。非可逆圧縮は、人間が知覚しにくい情報を取り除くことでデータ量を削減する手法である。一方、可逆圧縮は、原画像の情報を欠落させずに、データの冗長さを削減する手法である。非可逆圧縮は、一般的に、自然画像の圧縮に利用される。可逆圧縮は、医用画像 [3] や衛星画像 [4] といった、細部の情報が必要な用途に利用される。可逆圧縮は応用範囲が広いが、すべての画素を正確に記録する必要があるため、圧縮率が悪いという問題がある。

本稿では、複数の画像からなる画像集合を、可逆圧縮する手法について議論する。データセットの画像を 1 枚ずつ圧縮するのではなく、データセットの特徴を利用して、まとめて圧縮することで、より良い圧縮率が得られると考えられる。本稿では、最新の画像圧縮手法である JPEG XL [5] の可逆圧縮手法をベースとした、画像集合圧縮手法を提案する。

2 関連研究

2.1 画像圧縮技術

データ圧縮技術は、非可逆圧縮と可逆圧縮に分類でき、これは画像、動画、画像集合の圧縮符号化にも適用される。

画像の非可逆圧縮は、原画像を符号化し、復号化したときに得られる画像が、一般的に原画像と一致しない手法である。すなわち非可逆圧縮を行うことで、画像が劣化するといえる。一方、可逆圧縮は、原画像を符号化し、復号化したときに得られる画像が、原画像と完全に一致する手法である。一般的に、非可逆圧縮は可逆圧縮よりも圧縮率が良い。一般的な自然画像の符号化は、被写体を視覚的に違和感なく記憶できれば良いため、非可逆圧縮が用いられる。一方、画素ごとの値を記憶したい場合や、医用画像のように画像情報の損失が安全性に関わる場合には、可逆圧縮が用いられる [3]。可逆圧縮は、非可逆圧縮の用途にも適用可能である。

画像圧縮の主な手法として、変換符号化が挙げられる。変換符号化では、画像に対して、離散コサイン変換やウェーブレット変換といった周波数領域への変換を行うことで、画像を低周波成分と高周波成分に分離する。非可逆圧縮では、自然画像は高周波成分が視覚的に認知されにくい特性を利用し、高周波成分を粗く量子化することによって圧縮を行う。このような手法を用いる代表的な方式が JPEG や JPEG 2000 [6] である。また、量子化を行わないことで可逆圧縮となる。

可逆圧縮では、変換符号化の他に、画素値を直接符号化する手法も一般的である。画素値や、近傍画素を利用した予測値の誤差をエントロピー符号化することで、類似パターンに効率よく符号を割り当てることができる。このような手法を用いる代表的な方式が PNG である。

近年、画像圧縮に機械学習を用いる手法の研究が盛んになっ

ている．非可逆圧縮では，オートエンコーダを応用した，画像生成モデルを用いた手法が研究されている．[7]の手法は，BPG (HEVC ベースの画像圧縮手法) と比べて，半分のビットレートで同じ画質を実現した．可逆圧縮も，同様に機械学習を用いた手法が研究されている．Mentzer ら [8] は，オートエンコーダを可逆圧縮に応用した．オートエンコーダのデコーダ部は，画像そのものを予測するのではなく，画素値の確率分布を予測し，その確率分布を用いてエントロピー符号化を行う．Schiopu ら [9] は，1 画素ごとの予測を畳み込みニューラルネットワークで行う手法を提案した．しかし，Mentzer らの手法は，非ディープラーニングの手法の圧縮率を上回ることができなかった．Schiopu らの手法は，1 画素ごとに大量の計算が必要なため，符号化・復号化に時間がかかる問題がある．

2.2 動画圧縮技術

動画は，時間で標準化された画像列と考えることができる．動画は時間経過によって，カメラのパンやズーム，または被写体の移動によって画像が変化する特性を持つ．そこで，基準となる画像を画像圧縮技術で符号化し，その他のフレームについては基準画像からの変化量を符号化する．近年普及している動画符号化方式である AVC や HEVC [10] では，画像をいくつかのブロックに分割し，そのブロックが動いたと予測される向きや移動量と，それを使った予測画像との誤差を表す画像を，変換符号化を用いて符号化する．動画の非可逆符号化は，画像の非可逆符号化と同様に量子化間隔を変化させることで圧縮率を制御する．可逆符号化は，量子化を行わないことで実現される．

2.3 画像集合圧縮技術

画像集合圧縮技術とは，複数の画像を，画像間の冗長性を利用して，まとめて圧縮符号化する技術である．1 枚ごとに符号化を行うよりも圧縮率が良くなることを期待している．

画像集合の例として，同一被写体について，撮影した時刻やカメラの位置が異なる画像の集合を考える．このような画像集合は，類似する画像列であるという点で動画のような特性を持っているが，一方で，(1) 画像集合の要素を画像列として並べる方法がひとつではない，(2) 画像集合内の画像間の差異は，一般的に動画の 1 フレーム間の差異より大きい，という特性がある．

画像集合の圧縮は，代表信号ベースの手法と，グラフベースの手法の，大きく 2 種類に分類できる [11]．

代表信号ベースの手法は，画像集合内の画像の輝度値の平均や最大値・最小値を求めて作成した代表画像を利用する手法である．代表画像と，各画像の代表信号からの差を，それぞれ画像圧縮の手法で符号化する．この手法は，画像集合内の画像間の類似度が高く，被写体の移動が少ない画像集合に対して有効である．

グラフベースの手法は，画像間の距離 (差分を符号化するコスト) を定義し，画像を頂点，距離を辺の重みとするグラフを構築し，そのグラフを利用する手法である．このようなグラフは，重みの和が最小となるような木，すなわち最小全域木に変

換できる．重みの定義より，最小全域木を利用することで最適な画像列の並べ方を求めることができる．このようにして得られた画像列は，動画圧縮の手法を用いて符号化できる．グラフベースの手法は，距離の定義や画像列の符号化器の工夫によって，さまざまな画像集合に対応できる．グラフベースの手法は，平均画像を加えることで代表信号ベースの手法を内包できることが示されている [12]．

画像集合を可逆圧縮する手法としては，グラフベースの手法を利用し，最小全域木における親ノードからの差分画像を JPEG-LS で符号化する手法が提案されている [12]．

非可逆圧縮においては，近年では，距離に Scale Invariant Feature Transform (SIFT) を，符号化に HEVC を用いた手法が研究されている [13], [14]．SIFT で得られる特徴点を用いて，広域の座標・輝度変換を施した参照画像を作成し，符号化を行うことで，自然画像の集合の非可逆圧縮において高い圧縮率を達成している．

このような研究が行われていることを踏まえ，我々は，グラフベースの手法を応用し，動画符号化器としてアニメーション FLIF [15] を利用した画像集合の可逆圧縮手法を提案した [16]．この手法は，HEVC の可逆モードと比較して 7%以上の圧縮率の改善が見られた．しかし，ひとつのアニメーション FLIF ファイルに多くの画像が記録されてしまい，復号化に時間を要するという問題があった．また，FLIF はすでに開発が停止しており，FLIF の開発者らは後継となる JPEG XL の開発に取り組んでいる．そこで，本稿では，JPEG XL を利用した画像集合の可逆圧縮手法を議論する．

2.4 JPEG XL

JPEG XL [5] は，現在 ISO/IEC 18181 として標準化作業が進められている，画像の非可逆および可逆圧縮方式である．JPEG XL は，幅広いユースケースを想定しつつ，JPEG からの少ないコストで移行できるフォーマットを目指している．

JPEG XL には，非可逆，可逆，JPEG 再圧縮の 3 つの符号化モードがある．非可逆モードでは，ブロックサイズを可変とした離散コサイン変換 (DCT) を行い，JPEG のように量子化，エントロピー符号化を行う．可逆モードでは，画像から決定木として予測器の選択方法や予測誤差の傾向を学習し，その決定木を利用して符号化を行う．JPEG 再圧縮モードでは，入力 JPEG 画像の DCT 結果をそのまま利用しながら，予測やエントロピー符号化を最新の手法で行うことで，入力画像を劣化させずに圧縮を行う．

本稿では，可逆モードに注目する．可逆モードで使用される符号化器は，モジュラー符号化器と呼ばれ，非可逆モードでも DC 成分の符号化に使用されている．JPEG XL の可逆モードの大きな特徴として，画像を決定木として学習し，符号化方法を条件に応じて変化させる点 (図 1) と，エントロピー符号化に ANS (Asymmetric Numeral Systems) と呼ばれる手法を拡張した rANS [17] を使用している点が挙げられる．これらは，合わせて MAANS (Meta Adaptive / Asymmetric Numeral Systems) と呼ばれている．画像を決定木として学習すること

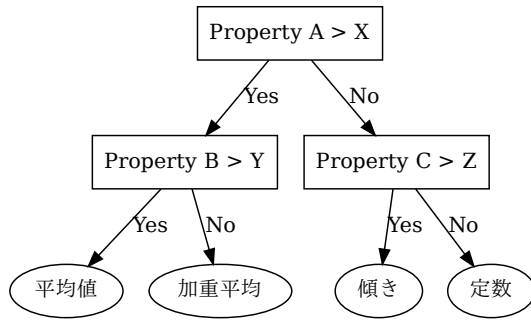


図 1 JPEG XL が学習する決定木のイメージ

で、複雑な形状を持つ人工的な画像に適応できる [18]。

3 提案手法

画像集合圧縮で主流となっているグラフベースの手法と、画像の可逆符号化で高い圧縮率を達成している JPEG XL を組み合わせ、画像集合を可逆符号化する手法を提案する。複数の画像をまとめて、JPEG XL 符号化器による決定木学習を行うことで、共通した特徴を抽出でき、決定木と画素の符号化における冗長性を削減できると考えられる。

本研究では、特定のテーマや被写体の自然画像や CG 画像の集合を対象とする。画像は、RGB 画像とし、集合に含まれるすべての画像の大きさは等しいと仮定する。

提案手法では、次の手順で画像集合の符号化を行う。まず、画像の特徴を抽出し、クラスタリングを行う。少ない計算コストで得られる特徴量を利用してクラスタリングを行うことで、符号化にかかる時間を削減できる。また、圧縮後、ランダムアクセスして復号化するときにかかる時間を予測可能にする。次に、クラスタ内でグラフベースの手法を用いて、最適な画像どうしの参照関係を推定する。最後に、得られた参照関係を利用して、JPEG XL による符号化を行う。提案手法の流れを図 2 に示す。本章では、各手順について説明する。

3.1 大まかなクラスタリング

はじめに、画像の特徴を抽出し、クラスタリングを行う。次節以降の処理で、画像の枚数の二乗に比例する回数の計算を行うため、事前にクラスタリングを行うことで、計算量を減らすことが目的である。また、これには、復号化時に同時に復号化する必要がある画像数の上限を設定し、復号化にかかる計算量を予測可能にする効果もある。

特徴量には、JPEG XL モジュラー符号化器が決定木学習に使用するプロパティ値（近傍画素の値や傾き）を利用する。まず、画像内での位置による傾向を得るため、画像を s 個の領域に分割する。分割された各領域で、 p 個のプロパティの平均と分散を求める。このようにして得られた $2ps$ 次元のベクトルを、画像の特徴量とする。以上の特徴量を使用することで、類似した決定木が学習されるであろう画像を同じクラスタにまとめることができる。

次に、得られた特徴量を用いてクラスタリングを行う。クラ

スタリングの目的から、1 クラスタあたりの画像数に大きなばらつきがないことが望ましい。そこで、クラスタに属する個体数（クラスタサイズ）の上限 \bar{K} と下限 \underline{K} を設定できる、COCBO 法 [19] を用いてクラスタリングを行う。COCBO 法は、k-means 法を改変した手法である。COCBO 法では、ある個体がどのクラスタに属するかを、 \bar{K} 、 \underline{K} を制約とする線形計画問題として計算する。

以上の手順でクラスタリングを行い、各クラスタについて、以降に示す手順で圧縮を行う。

3.2 符号化コストによる参照関係の推定

次に、グラフベースの画像集合圧縮手法を応用して、最適な圧縮方法を推定する。グラフベースの手法では、画像を頂点とし、ある画像を参照して他の画像を符号化するときのコストを重みとするグラフから、最小全域木を求めることで、最適な参照関係を得ることができる。最適な参照関係を求めるには、コストの定義が重要である。画像 I_A を参照して I_B を符号化するときのコスト $w(I_A, I_B)$ は、 I_A を参照して I_B を圧縮したときのビット数と正の相関があると良いと考えられる。

コストとして、他の画像の符号化状況を利用して、画像を符号化したときのビット数を考える。もし、ふたつの画像が類似しているならば、モジュラー符号化器が学習する決定木は類似し、その決定木を利用して符号化したときのビット数は小さくなる。そこで、画像 I_A を参照して画像 I_B を符号化するときのコスト $w(I_A, I_B)$ を、 I_A で学習した決定木を利用して I_B を符号化したときのビット数と、 I_B の通常のビット数との差とする。このような方法でコストを計算することで、符号化器にとっての画像の類似度を得ることができる。

このコストを用いたグラフから、最小全域木を求める。コストの定義より、一般的に $w(I_A, I_B) = w(I_B, I_A)$ は成り立たないので、このグラフは重み付き対称有向グラフとなる。有向グラフの最小全域木は、Chu-Liu/Edmonds アルゴリズム [20] で求めることができる。

最小全域木を求めるには、根となる画像をひとつ決める必要がある。ここでは、通常の JPEG XL で符号化したときにもっともビット数が少なくなる画像を根とする。

3.3 JPEG XL を利用した複数画像の符号化（画像列法）

最後に、前節の手順で得られた最小全域木を利用して、モジュラー符号化器による符号化を行う。画像は YCoCg 色空間へ変換されているものとする。 N 枚の画像をひとつの画像と見立て、 $3N$ チャンルの画像として、モジュラー符号化器で符号化を行う。複数の画像を同時に符号化することで、より特徴を捉えた決定木を学習しつつも決定木の冗長性を減らす効果があると考えられる。また、類似画像を符号化しているので、ANS 符号化時にシンボルの分布が偏りやすくなり、シンボルの出現確率を表すヒストグラムの符号化が効率的になると考えられる。

画像列法では、最小全域木の N 個の頂点に対応する画像を行きかけ順に並べた $3N$ チャンルの画像を作成し、モジュラー符号化器で符号化する。モジュラー符号化器ではチャンネルごと

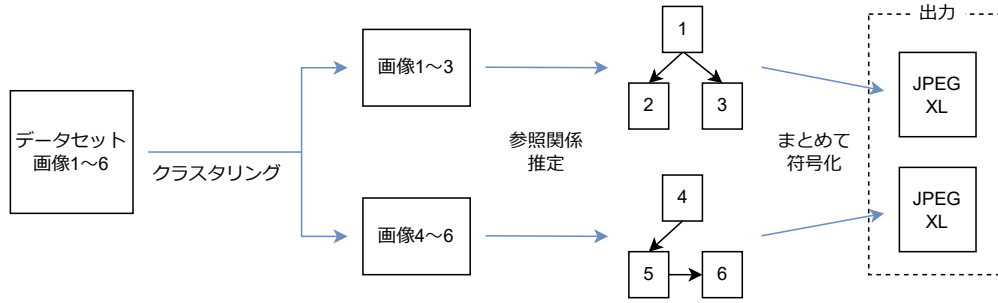


図2 提案手法

に符号化するので、チャンネル c を復号化するときチャンネル $c-1$ までは復号化済みなため、 $c-1$ までのチャンネルの画素値や符号化状況を参照できる。画像を行きがけ順に並べることで、ある画像の前にその親画像が符号化されるので、親画像を参照できるようになる。

3.4 最適な画像列を総当たりで探索（総当たり法）

画像列法では、クラスタ内のすべての画像を同時に符号化したが、同時に符号化しないほうが良い画像の組み合わせがある可能性がある。そのため、複数の画像を同時に符号化したときに、実際に圧縮率が良くなる場合だけ同時に符号化したい。そこで、最小全域木を探索しながら、良い結果が得られる組み合わせを選ぶ。次に頂点 v に対する探索アルゴリズム（帰りがけ順探索）を示す。次のアルゴリズムを根に対して適用することで、圧縮率の良い画像の組み合わせを得ることができると考えられる。

- (1) すべての子に、このアルゴリズムを適用する。
- (2) v に対応する画像を I_v とする。
- (3) v から出る辺を重みの小さい順にソートし、それぞれの辺の端 v_c について次の処理を行う。
 - (a) v_c に対応する画像を I_c とする。
 - (b) I_v に I_c のチャンネルを連結した画像を I_x とする。
 - (c) I_v, I_c, I_x を符号化したときのビット数 L_v, L_c, L_x を求める。
 - (d) もし $L_x < L_v + L_c$ ならば、 I_v に I_x を代入する。そうでないならば、 I_c を出力する。
- (4) もし v が根ならば、 I_v を出力する。

3.5 親画像を参照するプロパティの追加（親参照法）

画像列法では、チャンネル c を符号化するときチャンネル $c-1$ までを参照できることを利用した。しかし、オリジナルのモジュラー符号化器の他チャンネル参照は、親画像を参照するという目的に利用するには課題がある。そこで、モジュラー符号化器を拡張することを提案する。

オリジナルのモジュラー符号化器には、復号化済みのチャンネルの画素値や予測誤差を表すプロパティがある。しかし、ひとつ前のチャンネルに関するプロパティがプロパティ番号 16–19、ふたつ前のチャンネルに関するプロパティがプロパティ番号 20–23 といったように、現在のチャンネルからの相対的なプロパティ番号になっている。プロパティ番号は決定木の条件となる。した

がって、オリジナルの実装では「ある画像の Y チャンネルを参照」のようなチャンネルの絶対的な参照を行う条件が指定できない。そこで、プロパティ番号 16–19 は親画像の Y チャンネルに関するプロパティ、プロパティ番号 20–23 は親画像の C_0 チャンネルに関するプロパティ、……となるようにプロパティ番号を割り当てることを提案する。このようにプロパティ番号を割り当てることで、親画像で特徴的だった部分を利用して適切な予測ができるようになると思われる。

4 評価

4.1 圧縮率の比較

提案手法を評価するために、3 種類のベースライン手法と 3 種類の提案手法でデータセットを圧縮し、圧縮結果のバイト数で比較を行う。ベースライン手法は以下の通りである。

(1) JPEG XL 1 枚ずつ JPEG XL で可逆圧縮を行う。この実験によって、提案手法が画像集合の特性を利用して圧縮率を改善するかを評価する。

(2) Anim FLIF 先行研究である画像集合圧縮にアニメーション FLIF を利用する手法 [16]。この実験によって、JPEG XL を用いることで、他の符号化手法を利用するよりも圧縮率が改善するかを評価する。

(3) 平均差分 各クラスタの圧縮に、平均画像を代表信号とする差分画像圧縮を適用する手法。この実験によって、提案手法がもっとも単純な画像集合圧縮の手法よりも圧縮率が改善するかを評価する。

実験する提案手法は、以下の通りである。

(1) 画像列法 第 3.3 節の手法。直前 3 チャンネルを参照しながら符号化する。

(2) 総当たり法 第 3.4 節の手法。同様に、直前 3 チャンネルを参照しながら符号化する。

(3) 親参照法 第 3.5 節の手法。親画像の 3 チャンネルを参照しながら符号化する。

実験は、Windows 10 上の WSL2 で行う。使用するハードウェアは一般的な PC で、CPU は Intel Core i7-9700 (8 コア)、メモリは 16 GB (最大 9 GB を WSL2 に割り当て) である。使用するソフトウェアのバージョンを表 1 に示す。FLIF 符号化器および JPEG XL 符号化器は、提案手法を実装するため、表 1 のバージョンをベースに改変を行った。JPEG XL の圧縮レ

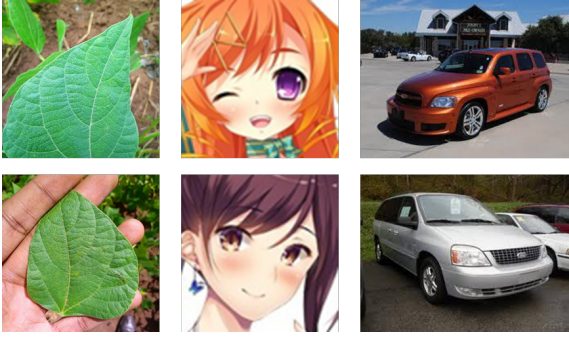


図3 実験に使用するデータセット．左から ibean , Anime Face , Cars

ベルは、もっとも高い tortoise とする．クラスタリングに使用する，クラスタごとの画像数の制約は， $K = 20$ ， $\bar{K} = 17$ ， $\bar{K} = 24$ とする．特徴量抽出時の画像の領域分割数は $s = 4$ (上下左右に分割) とする．

画像集合には，インターネットで公開されている画像データセットを利用する．使用するデータは次の3つである(図3)．

(1) ibean¹ 豆の葉を撮影した 500×500 の実写画像．学習画像の healthy ラベル 341 枚を使用．

(2) Anime Face² ゲーム・アニメに関する Web サイトから収集した CG 画像．2019 年の作品に関する画像から重複を取り除いた 1389 枚を 95×95 にリサイズしたものを使用．

(3) Cars [21] 自動車を撮影した実写画像．テスト用画像から大きさ 259×194 の画像 423 枚を抽出．

各手法で圧縮を行ったときのデータ量を表2に示す．表2より，画像列法は，画像を1枚ずつ JPEG XL で圧縮した場合に対して，Anime Face で 1.1%，Cars で 2.6% 圧縮率が良かった．一方で，ibean では，1枚ずつ圧縮したほうが圧縮率が良かった．また，いずれのデータセットでも，提案手法は，他の手法 (Anim FLIF，平均差分) よりも良い結果が得られた．提案手法の中では，Anime Face と Cars においては画像列法が，ibean においては総当たり法が最も圧縮率が良かった．

4.2 クラスタサイズによる符号化・復号化時間の変化

提案手法では，符号化および復号化にかかる時間を予測可能にするために事前にクラスタリングを行った．そこで，実際にクラスタリングのパラメータによって，符号化・復号化時間が抑制されるかを確認する．

クラスタサイズ K ($\underline{K} \leq K, K+1 \leq \bar{K}$) を小さくすると，クラスタ数は増加し，1 クラスタあたりの画像数が少なくなる．1 クラスタあたりの画像数が少ないと，圧縮率は悪化するが，ひとつの画像を符号化・復号化するために必要な計算量が少なくなるので，符号化・復号化時間は短くなると考えられる．

Anime Face データセットを対象として，実験を行う．まず，クラスタサイズ K を変化させたながらデータセットを提案手法で符号化し，その時間を計測する．次に，圧縮結果からランダムに1枚を復号化する操作を1000回行い，その時間を計測

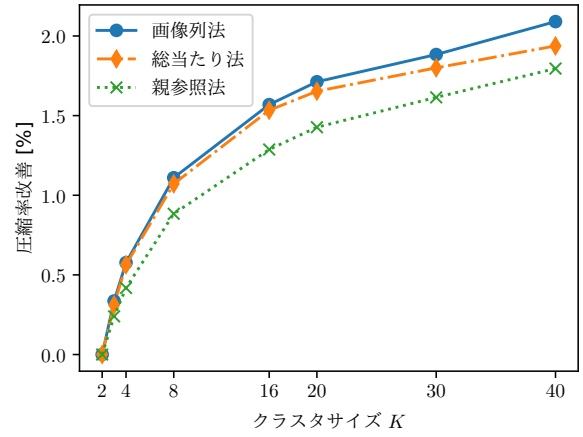


図4 クラスタサイズに対する圧縮率の変化 ($K = 2$ 基準)

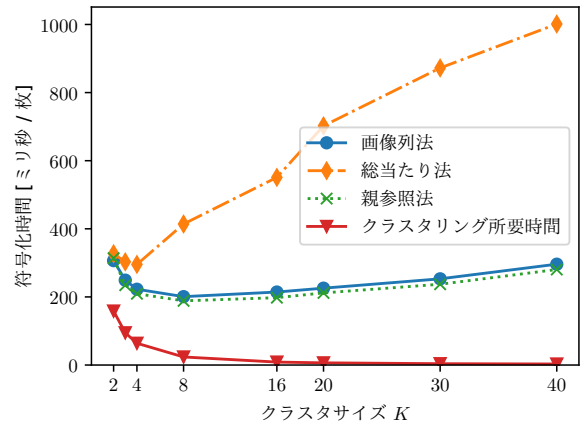


図5 クラスタサイズに対する1枚あたりの平均符号化時間の変化

する．クラスタリングのパラメータは $\underline{K} = K$ ， $\bar{K} = K + 1$ とする．

クラスタサイズ K を変化させたながらデータセットを符号化したときの，圧縮率の変化 ($K = 2$ のときを基準に何%バイト数を削減できたか) を図4に，1枚あたりの平均符号化時間を図5に示す．図5の「クラスタリング所要時間」は画像列法での符号化のうちクラスタリングに要した時間を表している．図4より，いずれの手法も圧縮率は K を大きくするほど改善することがわかる．図5より，所要時間は，画像列法と親参照法では， $K = 2$ のときに長く，その後 $K = 8$ で極小を迎えることがわかる．総当たり法では， $K = 4$ で極小を迎え，その後は所要時間が大きく増加することがわかる．符号化時間に占めるクラスタリングの時間は，クラスタサイズが大きいくほど短くなる．次に，1枚あたりの平均復号化時間を図6に示す．画像列法と親参照法では，復号化時間は K に比例することがわかる．総当たり法では，復号化時間は画像列法と親参照法よりも短くなる．

4.3 決定木とエントロピー符号の調査

JPEG XL で1枚ずつ圧縮したときと，提案手法で圧縮したときの圧縮結果の違いを調査するため，エントロピー符号化の傾向を求める．JPEG XL では，エントロピー符号を復号化する

1: <https://github.com/AI-Lab-Makerere/ibean>

2: <https://www.kaggle.com/splcher/animefacedataset>

表 1 実験環境のソフトウェア

ソフトウェア名		バージョン
ホスト OS	Windows 10 Pro (64bit)	21H2
ゲスト OS	Ubuntu	20.04.3 LTS
FLIF 符号化器	FLIF ³	0.4
JPEG XL 符号化器	libjxl	2021 年 12 月 15 日現在のソースコード ⁴
最適化問題ソルバ	libglpk	4.65

³ <https://github.com/FLIF-hub/FLIF>⁴ <https://github.com/libjxl/libjxl/tree/3d077b281fa65eab595447ae38ba9efc385ba03e>

表 2 各手法での圧縮結果 [MB]

データセット	原画像	ベースライン手法			提案手法		
	PNG	JPEG XL	Anim FLIF	平均差分	画像列法	総当たり法	親参照法
ibean	192.1	104.9	111.7	119.4	105.7	105.5	108.1
Anime Face	26.26	16.44	17.21	19.76	16.22	16.23	16.38
Cars	37.14	17.14	18.30	27.40	16.67	16.72	17.07

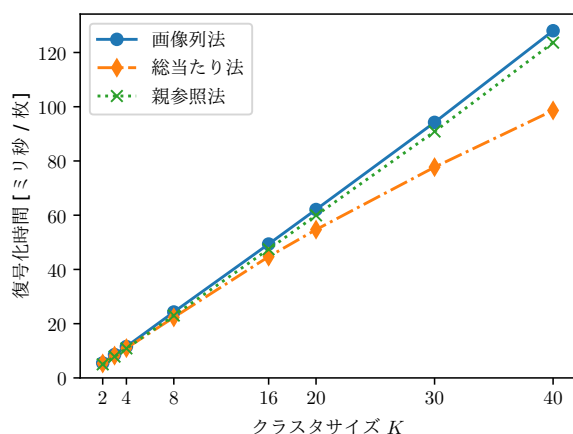


図 6 クラスサイズに対する 1 枚あたりの平均復号化時間の変化

るために、学習した決定木と、ANS での各シンボルの出現頻度 (rANS [17] の f_s) を出力に含めている。そこで、JPEG XL で 1 枚ずつ圧縮したとき、提案手法 (圧縮率が良かった画像列法を使用する) で圧縮したときの圧縮結果を解析し、決定木とシンボルのヒストグラムに使用されているビット数を比較する。

第 4.1 節で実験した JPEG XL、画像列法の圧縮結果から、決定木とシンボルのヒストグラムに使用されているビット数の合計を求めた。画像列法におけるそれぞれのビット数と、JPEG XL との差を、表 3 に示す。実験結果から、画像列法ではヒストグラムに要するビット数を約半分に削減できることがわかった。一方、決定木のビット数は、JPEG XL よりも圧縮率が良かった Anime Face と Cars では減少し、圧縮率が悪かった ibean では増加した。

次に、直前 3 チャンネルを参照する画像列法と、親参照法の特徴の違いを調査するため、他チャンネルの参照状況を調査する。他チャンネル参照が決定木の条件としてどの程度利用されているかを調べる。圧縮結果の決定木から、葉以外の頂点について、条件となっているプロパティを集計する。該当するすべての頂

表 3 画像列法で圧縮したときの決定木とシンボルヒストグラムのビット数 [kbit]。括弧は 1 枚ずつ圧縮したときとの差。

データセット	決定木	ヒストグラム
ibean	5178 (+9.38%)	1896 (−64.4%)
Anime Face	897.6 (−24.8%)	975.7 (−46.7%)
Cars	1294 (−12.6%)	780.8 (−64.7%)

表 4 他チャンネルを参照するプロパティが使用された頂点の割合 [%]

データセット	画像列法	親参照法
ibean	20.1	15.4
Anime Face	14.0	4.30
Cars	13.4	6.57

点のうち、他チャンネルを参照するプロパティが使用されている割合を求める。

圧縮結果から求めた他チャンネルを参照するプロパティが使用割合を表 4 に示す。実験結果から、いずれのデータセットでも、親参照法よりも画像列法のほうが他チャンネルを参照するプロパティの使用頻度が多いことがわかった。

4.4 考 察

実験結果 (表 2) より、JPEG XL を用いて画像を 1 枚ずつ符号化するより、提案手法のほうが圧縮率が改善するケースがあることがわかった。圧縮率が改善したケースでは、表 3 からわかるように、決定木とシンボルヒストグラムの冗長性を削減できたことで、圧縮率が改善したと考えられる。

一方で、ibean データセットを提案手法で符号化した場合、1 枚ずつ符号化するよりも圧縮率が悪くなった。これは、ibean が実験した他のデータセットよりも画素数が多いことが原因だと考えられる。画素数が多いと、画像間に類似した箇所があっても、位置の差が大きくなる (画素が遠くなる) ため、共通した特徴として利用しにくくなる。そのため、1 枚ずつ決定木を学習するよりも、画像の特徴を捉えにくくなったのではないかと

考えられる．また，色空間の選択にも違いがある．提案手法は，複数の画像をまとめるため YCoCg のみを利用したが，JPEG XL ではさまざまな可逆色変換を利用できる．ibeam は緑が多いため，緑を第 1 チャンネルと置く色変換を行うほうが，圧縮率が改善すると考えられる．

総当たり法は，最適な画像列が得られるだろうという仮説に反して，画像列法よりも圧縮率が悪いケースがあった．総当たり法は，圧縮率が改善する場合のみ，画像を結合する手法である．このアルゴリズムは木構造を参考に結合方法を選択しているので，取りうるすべての結合方法を試行するわけではない．実験結果から，提案した総当たり法のアルゴリズムでは，必ずしも圧縮率が改善する組み合わせを選択することはできないといえる．

親参照法は，画像列法より圧縮率が悪かった．この理由は，他の画像を参照するよりも，同じ画像の他チャンネルを参照するほうが画素値の予測の参考になるからだと考えられる．実際，表 4 より，親参照法は画像列法よりも他チャンネルを参照するプロパティの使用頻度が低い．これは，親参照法では決定木の学習時に，他チャンネルを参照することは有効ではないと判断されることが多かったといえる．

符号化・復号化にかかる時間は，実験から，画像列法と親参照法では復号化時間がクラスタサイズに比例することがわかった．このことから，目的どおり，クラスタサイズによって圧縮率と復号化時間のトレードオフを設定できたといえる．総当たり法では，結合された画像数がクラスタサイズよりも少なくなる場合があるので，画像列法と親参照法よりも復号化時間が短くなった．符号化時間は，クラスタサイズが小さいときに時間がかかり，クラスタサイズを増加させると極小に達し，さらに増加させると符号化時間は長くなっていくという結果が得られた．これは，クラスタサイズが小さいとき，クラスタリングにかかる時間が支配的になり，クラスタサイズが大きいき符号化にかかる時間が支配的になるからである．特に総当たり法は，クラスタ内のさまざまな組み合わせを試して，圧縮結果が最小になる符号化方法を探索するので，クラスタサイズが多いと計算量が多くなる．一方で，クラスタサイズを小さくするとクラスタ数が多くなるので，COCBO 法で計算する線形計画問題の変数と制約条件が多くなり，計算が収束するまでに時間がかかる．したがって，符号化時間を短くするには，線形計画法が速く収束するようなクラスタサイズを選択する必要があることがわかる．

5 ま と め

JPEG XL を応用した新しい画像集合の可逆圧縮手法を提案した．提案手法では，符号化器の特性を表す特徴量を利用したクラスタリングを行い，クラスタに含まれる画像のチャンネルを結合して符号化を行う．実験から，提案手法は，従来の平均画像を利用した圧縮手法や，FLIF を利用した圧縮手法よりも圧縮率が良いことがわかった．一方で，画素数の多い自然画像においては，オリジナルの JPEG XL を利用して 1 枚ずつ圧縮し

たほうが圧縮率が良いことがわかった．また，クラスタリングのパラメータによって，圧縮率・復号化時間のトレードオフを設定できることを示した．

今後の課題として，動画符号化で用いられている手法を導入が挙げられる．実験結果より，他の画像を参照するプロパティの追加は，効果がなかった．他の画像を有効に参照するには，被写体の移動や，明るさの変化などを考慮する必要があると考えられる．動画符号化で用いられている手法を導入することで，他の画像を有効に利用でき，圧縮率が改善すると考えられる．また，提案手法は従来手法に比べて符号化に時間がかかるので，高速化が課題である．

文 献

- [1] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 843–852, 2017.
- [2] Olga Russakovsky, Jia Deng, Hao Su, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, Vol. 115, No. 3, pp. 211–252, 2015.
- [3] S. Bhavani and K. Thanushkodi. A Survey On Coding Algorithms In Medical Image Compression. *International Journal on Computer Science and Engineering*, Vol. 2, No. 5, pp. 1429–1434, 2010.
- [4] Vivek Trivedi and Howard Cheng. Lossless compression of satellite image sets using spatial area overlap compensation. In Mohamed Kamel and Aurélio Campilho, editors, *Image Analysis and Recognition*, pp. 243–252, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [5] Jyrki Alakuijala, Ruud van Asseldonk, Sami Boukortt, et al. JPEG XL next-generation image compression architecture and coding tools. In *Applications of Digital Image Processing XLII*, Vol. 11137, pp. 112–124. SPIE, 2019.
- [6] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, Vol. 18, No. 5, pp. 36–58, 2001.
- [7] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. In *Advances in Neural Information Processing Systems*, Vol. 33, pp. 11913–11924. Curran Associates, Inc., 2020.
- [8] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, et al. Practical full resolution learned lossless image compression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10621–10630, 2019.
- [9] Ionut Schiopu and Adrian Munteanu. Deep-learning-based lossless image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 30, No. 7, pp. 1829–1842, 2020.
- [10] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1649–1668, 2012.
- [11] Z. Bohari and M. S. Raen. Overview of State-of-the-Art Image Set Compression Techniques. *International Journal of Science and Research (IJSR)*, Vol. 4, No. 4, pp. 709–712, 2015.
- [12] Barry Gergel. *Automatic compression for image sets using a graph theoretical framework*. PhD thesis, University of Lethbridge, 2007.
- [13] Zhongbo Shi, Xiaoyan Sun, and Feng Wu. Photo album

- compression for cloud storage using local features. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 4, No. 1, pp. 17–28, 2014.
- [14] Lina Sha, Wei Wu, and Bingbing Li. Novel image set compression algorithm using rate-distortion optimized multiple reference image selection. *IEEE Access*, Vol. 6, pp. 66903–66913, 2018.
 - [15] Jon Sneyers and Pieter Wuille. FLIF: Free lossless image format based on MANIAC compression. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 66–70, 2016.
 - [16] Takuya Sakurai and Ushio Inoue. Lossless image set compression using animated FLIF. In *8th International Virtual Conference on Applied Computing & Information Technology (ACIT)*, pp. 99–104. ACM, 2021.
 - [17] Jarek Duda, Khalid Tahboub, Neeraj J. Gadgil, and Edward J. Delp. The use of asymmetric numeral systems as an accurate replacement for huffman coding. In *2015 Picture Coding Symposium (PCS)*, pp. 65–69, 2015.
 - [18] Jon Sneyers. Compressing cellular automata, 2017. https://cloudinary.com/blog/compressing_cellular_automata.
 - [19] Yasunori Endo, Sachiko Ishida, and Naohiko Kinoshita. Controlled-sized clustering based on optimization. In *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, pp. 1–6, 2017.
 - [20] Harold N Gabow, Zvi Galil, Thomas Spencer, and Robert E Tarjan. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, Vol. 6, No. 2, pp. 109–122, 1986.
 - [21] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops, ICCVW '13*, pp. 554–561, USA, 2013. IEEE Computer Society.