論　文

マーカコードの発生を考慮したJPEG2000符号化画像の情報半開示法

貴家　仁志[†]　　今泉　祥子[†]　　渡邊　修[†]

Partial-Scrambling of JPEG2000 Images without Generating Marker Codes

Hitoshi KIYA[†], Shoko IMAIZUMI[†], and Osamu WATANABE[†]

あらまし　本論文では，特殊命令コードであるマーカコードを発生せず，かつ新しい制御パラメータを有するJPEG2000符号化画像のための情報半開示方式を提案する．制御パラメータの導入は，半開示の画質，演算量及び不正解除に対する強度に関する制御能力を向上させる．演算量に関する制御能力の向上は，サイズの異なる画像に対して，半開示処理に要する処理負担をほぼ一定にすることを可能とする．この特長は，動画像のリアルタイム処理において重要な役割を果たす．

キーワード　情報半開示，JPEG2000，マーカコード，EBCOT，画像圧縮

1. まえがき

近年，コンピュータの性能向上，ネットワークの高帯域化により，ディジタル画像の扱いが容易となってきた．これに伴い，ネットワークやCD-ROMを介した，商品としてのディジタル画像の流通が増加してきており，その著作権保護が問題となっている．ディジタルデータが品質の劣化なくコピーできるためである．一方で，医療分野でディジタル画像を利用する機会も増えており，このとき，画像のプライバシーの保護が問題となる．これらの問題に対し，ディジタル画像の著作権保護，プライバシー保護の手法として，一般に三つの種類のものが存在する．すなわち，(a) 電子透かし [1], [2]，(b) 画像情報全体の暗号化 [3]，(c) 情報半開示 [4]〜[11] である．本論文では，JPEG2000 符号化画像に対する (c) の半開示方式を提案するものである．電子透かしは，著作権情報を画像に埋め込むことにより，不正利用を抑止する方式である．しかし，この方式では，画像の不正利用や不正コピー自体を未然に防止することが困難である．更に，画像を使用する人，画像を写された人のプライバシーを保護することもできない．一方，画像情報全体の暗号化は，画像情報の保護のための方式として最も堅牢であるが，完全に復号しなければ画像の内容を確認することができず，画像データの扱いや検索に制約が大きい．

そこで，商品として画像を流通させる場合の著作権保護，また医用画像等のプライバシー保護に有用な方式として，画像の情報半開示が期待されている．これは，画像をスクランブルし，その概要が確認できる程度に品質を劣化させる方式である．この方式では，画像提供者はあらかじめスクランブルした画像をネットワークやCD-ROMなどの媒体を通して配布する．一方，画像利用者は，スクランブルされた画像を見てその概要を確認し，必要に応じて画像提供者から復号鍵とパラメータ情報を入手して，スクランブルを解除する．JPEG2000 の標準化作業においても，Part8 では，この情報半開示法がセキュリティ機能の一つとして検討されている [12], [13].

情報半開示の方法は，画像を直接スクランブルする方式 [3] と，画像圧縮を前提にする方式に大別される．JPEG2000 符号化画像の情報半開示法として既に提案されている従来方式 [6]〜[11] は，後者に属す．しかしながら，従来法には二つの大きな問題が伴う．まず，一つはマーカコード発生の可能性である [6]〜[10]．スクランブル処理の際，コードストリームの一部を無作為に他の値に写像するため，スクランブル処理によりマーカコードが発生する可能性がある．マーカコードとは特別な意味をもつ命令コードであり，一般にこのコードが発生すると正常な復号が保証されない．したがって，従来法ではスクランブル処理の後にマーカコードの発生を確認し，もし発生が確認された場合に

[†] 東京都立大学大学院工学研究科電気工学専攻，八王子市
Electrical Engineering, Tokyo Metropolitan University, 1-1 Minamiosawa, Hachioji-shi, 192–0397 Japan

は再度スクランブル処理を実行し直す必要があった．他の一つは，画質，演算量及び不正解除に対する強度に関する制御自由度が不十分であることである [6]〜[11]．従来法では，これらの制御はスクランブル対象領域（レイヤ，解像度レベル及びコードブロック）の選択のみで行っていた．このことは処理時間の制御の問題にも影響する．動画像のように実時間処理が強く求められる応用において，処理時間が画像サイズに依存するため，ほぼ一定の処理時間を保証することができない．

このような背景から，本論文では，従来法の特長を保持して，問題点を解決したスクランブル処理法を提案する．提案法は，制約付き半バイト単位のスクランブル処理及び制御パラメータの導入により，上述した問題点を解決するものである．また，シミュレーションによりその効果を確認している．

2. JPEG2000 の概要と従来法

ここでは，JPEG2000 符号化方式 [14], [15]，スクランブル処理において求められる条件とその従来方式 [6]〜[11]，及び JPEG2000 におけるマーカコード [14], [15] について要約し，本研究の課題を示す．

2.1 JPEG2000 の符号化方式

図 1 は，JPEG2000 エンコーダのブロック図である．入力画像は，まず，ウェーブレット変換 (DWT) によりサブバンド分解され，その後量子化される．量子化されたウェーブレット係数は，EBCOT (Embedded Block Coding with Optimized Truncation) [16] アルゴリズムにより符号化される．提案する半開示方式は，この EBCOT の機能を利用する．EBCOT は，コードブロック分割，係数モデリング，算術符号化とレート制御，レイヤ形成，パケット生成の五つの部分に分けることができる．

図 2 は JPEG2000 コードストリームの構造の一例である [15]．同図は，コードストリームがレイヤ構造をとった場合に対応している．この場合，コードストリームは，グローバルヘッダに最上位から順にレイヤが続く形で構成され，2 バイトの一つのマーカである EOC (end of code-stream) で終点となる．グローバルヘッダはコードストリーム全体を再生するために必要な包括的な情報を含んでいる．個々のレイヤは，一連のパケット群である圧縮されたパックストリームからなる．各パケットはヘッダとボディとからなり，レベル 0 を除き，サブバンドごとに分解され情報が入れられている．パケット単位は解像度レベルごとで，レベル 0 のパケットにはサブバンド分解によって生成された低域 (LL) 成分の情報のみが，その他のレベルには LL 以外の三つの成分 (HL，LH，HH) がそれぞれ含まれる．

従来法 [6]〜[11] は，レイヤ，解像度レベル，またはコードブロックというスクランブルの処理単位を選択し，いずれもそのボディデータに対してスクランブル処理を施している．これらの点において，提案法も同様である．

2.2 スクランブル処理に対する要求条件

提案するスクランブル処理は次のような要求を満たすことを想定する．

(a) スクランブル処理及び解除は，コードストリームに対してのみ施される．

(b) スクランブル処理によって，コードストリームのサイズを変えてはならない．

(c) スクランブルされたコードストリームは，JPEG2000 Part1 [14] に準拠しなければならない．

(d) スクランブル処理法は，スクランブルの程度及び強度を制御できる．

(e) スクランブル処理法は，スクランブルの演算量を制御できる．

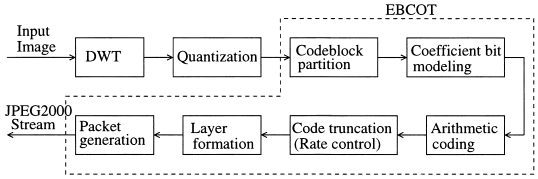


図 1　JPEG2000 エンコーダ
Fig. 1　JPEG2000 encoder.

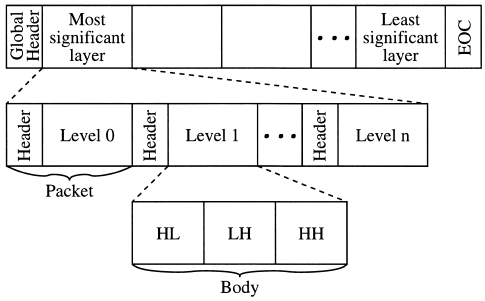
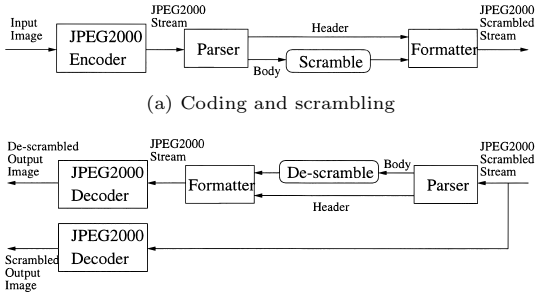


図 2　JPEG2000 コードストリームのレイヤ構成
Fig. 2　The layer-structure of JPEG2000 code-stream.



(a) Coding and scrambling

(b) De-coding and de-scrambling

図 3　JPEG2000 符号化とスクランブル処理
Fig. 3　Scrambling/de-scrambling with JPEG2000 codec.

JPEG2000 符号化のスクランブル処理とその解除は図 3 のように実行される．汎用エンコーダの出力である JPEG2000 ストリームは，構文解析器 (Parser) によってヘッダとボディに分解されボディのみにスクランブル処理が施される（上記の要求条件 (a)）．また，スクランブル処理されたコードストリームは，レート制御とヘッダ情報に影響を与えないためにスクランブル処理前のデータ量と同じデータ量となる（要求条件 (b)）．その後，構文合成器 (Formatter) で JPEG2000 ストリームとして再構成され，符号化データとなる．復号化側では，符号化側における逆処理によってスクランブルが解除され，解除された JPEG2000 ストリームが汎用デコーダに送られ再生画像が生成される（要求条件 (a)）．一方，復号側においてスクランブルの解除を行わず，符号化側でスクランブルされた JPEG2000 ストリームが汎用デコーダに送られた場合，その出力として半開示画像が生成される（要求条件 (c)）．

JPEG2000 符号化画像における情報半開示法の従来法 [6]〜[10] では，レイヤ，解像度レベルまたはコードブロック単位で画像を直接表すボディデータに対してスクランブル処理を実行している（要求条件 (a), (b)）．また，レイヤ，解像度レベル及びコードブロックを選択することにより，スクランブルの程度，強度及び演算量を制御する（要求条件 (d), (e)）．しかし，スクランブル時，ボディデータを 1 バイト単位で任意の値に写像するため，マーカコードの発生の可能性がある．マーカコードが発生した場合，要求条件 (c) を満たすことはできない．マーカコードは，JPEG2000 において特別な意味を保つ符号であり，スクランブル処理により偽のマーカコードが生成されると，半開示画像の正常な再生が妨げられることがある．したがって，従来法は要求条件 (c) を常に満たすことは困難である．更に従来法は，後述するように，要求条件 (d) 及び (e) の制御能力も十分ではない．提案法は，従来法の特長を保持したまま，従来法における要求条件 (c), (d) 及び (e) の問題を改善するものである．

2.3　JPEG2000 におけるマーカコード

本論文で扱うマーカコードとは，$FF90_h - FFFF_h$ の値を伴う特殊命令コードである．ここで，下付き h は 16 進数の記述を意味する．マーカコードは 2 バイトで表され，先頭の 1 バイトは FF_h である．更に用途に応じて $FFxx_h$ のように 1 バイトの情報 xx_h が付加される．

JPEG2000 では，一般にマーカコードは圧縮データ自身（図 2 のパケットボディ）の中に存在しない．すなわち，算術符号化（MQ コーダ）は，これらのコードを発生しないように設計されている．本論文で回避したいのは，この 2 バイトを単位とする $FF90_h - FFFF_h$ のマーカコードである．

3.　提　案　法

従来法 [6]〜[10] は，レイヤ，解像度レベル及びコードブロックを選択し，1 バイト単位で構成されるボディデータにスクランブル処理を施すものである．提案法も基本的には従来と同様にこのボディデータにスクランブル処理を施すものである．提案法の目的は，従来法 [6]〜[10] のもつマーカコードの発生，画質，強度及び演算量の不十分な制御能力という問題点を改善することにある．

3.1　マーカコードの回避条件

2.3 で述べたように，2 バイトの値である $FF90_h - FFFF_h$ のマーカコードは，圧縮データ自身の中には本来発生しない．したがって，MQ コーダの出力であるそのボディデータのスクランブル処理は，この範囲のマーカを発生しないことが考慮されていなければいけない．

マーカコードが 2 バイトであるため，スクランブルのためにある 1 バイトを他の 1 バイトに写像する場合，一般にその 1 バイトの値のみならず前後の 1 バイトとの関係，すなわち連続する 3 バイトの値を調べる必要がある（図 4 参照）．しかし処理の単純化のために，ここでは，スクランブル処理前にマーカコードは存在しないことを前提に，前後関係を考慮せずにマーカコードの発生を回避可能な 1 バイトごとの写像条件

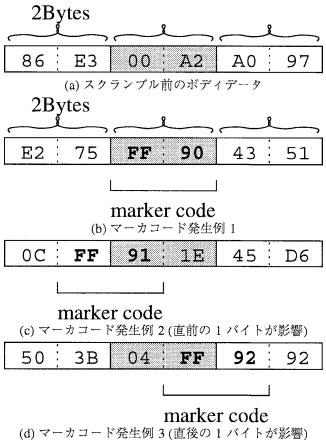


図 4　マーカコードの発生例とその種類
Fig. 4　The generation of marker-codes and its types.

を考察する．

まず，着目する 1 バイトの値を大きさにより三つに分類した次の条件を考える．

(a1) 着目する 1 バイトの値が 90_h 未満ならば，スクランブル処理後も 90_h 未満を保証すること

(a2) 着目する 1 バイトの値が 90_h 以上（FF_h を除く）ならば，スクランブル処理後も FF_h 未満を保証すること

(a3) 着目する 1 バイトの値が FF_h の場合，スクランブル処理後の値は任意の 1 バイトを選択可

ここで，(a1) は，直前と直後の 1 バイトがともに FF_h であってもマーカコードにならない条件である．(a2) は，直前の 1 バイトが FF_h であることはない場合の条件であり（もし FF_h ならば既にマーカコードであるので），スクランブル処理後も直前が FF_h でなければ，直後の関係も含めマーカコードになることはない．(a3) は，直前の 1 バイトが FF_h ではなく，かつ直後の 1 バイトが 90_h 未満の場合の条件であり，スクランブル処理後もそれらの条件を満たすので ((a1)(a2) より)，この場合には制約は必要ない．このような条件が満たされない場合でも，マーカコードが発生するとは限らないことに注意してほしい．例えば，(a1) を満たさない場合でも，その直前の 1 バイトが FF_h ではなく，かつ直後の 1 バイトが 90_h 未満になる可能性があるからである．したがって，上述の条件はマーカコード回避のための十分条件となる．

従来法のように，各 1 バイトデータを単に任意の 1 バイト値に写像するスクランブル法では，上述のマーカコード回避条件を満たすことはできない．上述の条件を満たすために，1 バイトを上位半バイトと下位半バイトに分割し，スクランブル対象をどちらかの半バイトに限定する方法を提案する．このとき，上述の条件にそれぞれ対応する以下の条件を考えることができる．

(b1) 着目する 1 バイトの値が 90_h 未満ならば，下位半バイトを選択すれば，スクランブル処理後の値は任意の半バイトを選択可

(b2) 着目する 1 バイトの値が 90_h 以上（FF_h を除く）ならば，その上位（または下位）半バイトを選択し，スクランブル処理後も F_h 未満を保証すること

(b3) 着目する 1 バイトの値が FF_h の場合，上位（または下位）の半バイトを選択すれば，スクランブル処理後の値は任意の半バイトを選択可

上述の条件 (b) は，条件 (a) に対して十分条件となっている．半バイト単位のスクランブル処理を考える理由は，暗号化アルゴリズム等を使用する際に必要となる任意の値への写像を許容可能にしたいからである．条件 (b2) を除き，条件 (b) では任意の値への写像が可能となっている．

以下では，条件を満たす具体的なスクランブル処理法について述べる．

3.2　スクランブル処理の手順

上述の条件 (b) を満たす具体的な方法を示す．従来法 [6]〜[10] との主な違いは，半バイト単位にデータを選択することによりマーカコードの発生を回避するとともに，パラメータ M により画質，強度及びスクランブル処理演算量に関する制御能力を向上させることにある．以下で具体的なスクランブル処理の手順を説明する．

3.2.1　スクランブル処理法

具体的なスクランブル処理法は，コードストリームのサイズを変えないことが考慮され，かつ一意にスクランブル解除可能である必要がある．ここでは，更に上述のマーカコードの回避条件を満たす二つの処理法を述べる．その一つは，1 バイト単位に異なる 1 バイトへの写像を行う暗号化アルゴリズム（例えば Blowfish [17]）の使用に基づく方法である [9]．他の一つは，半バイト単位のビットシフトを行うスクランブル処理である．暗号化アルゴリズムを使用した場合，スクランブル処理後の値は任意の値をとるのに対して，ビットシフトでは処理後の値は限定されたものになる．

はじめに，暗号化アルゴリズムの使用を前提にした

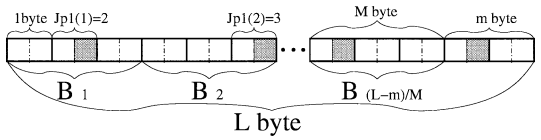


図 5　着目する L バイトのボディデータ（$M = 3$, $m = 2$ の場合）：網掛け部分をスクランブル
Fig. 5　Body data with L Bytes ($M = 3$, $m = 2$); hatched parts are scrambled half Bytes.

方法から述べる．まず，図 3 (a) に示すように，エンコーダから出力された JPEG2000 ストリームを構文解析器に送りヘッダとボディデータに分解する．次に，適当なレイヤ，サブバンド，またはコードブロックを選択し，そのボディデータ（L バイトと仮定する）を抽出する．その L バイトのボディデータを M バイトごとに区切り，各ブロックを B_k ($k = 1, 2, \ldots, (L-m)/M$, $m = \mathrm{mod}(L, M)$) とする．これらのデータを 1 バイト単位で考え，その各下位半バイトに対して次の処理を実行する（図 5 参照）.

（1） ある初期値 p_1 とある整数乱数発生アルゴリズム $\boldsymbol{J}_{p_1}(k)$ を用いて，1 から M までの整数乱数,

$$\boldsymbol{J}_{p_1}(k) \in \{1, 2, \ldots, M\},\ k = 1, 2, \ldots \tag{1}$$

を発生させる.

（2） M バイトの各 B_k において $\boldsymbol{J}_{p_1}(k)$ 番目の 1 バイトをそれぞれ選択する.

（3） 各ブロックごとに選択された 1 バイトに対し，次の条件に従ってスクランブル処理の対象となる半バイトをそれぞれ選ぶ.

（3.1）選択された 1 バイトの値が $F0_h$ 未満ならば，その下位半バイトをスクランブル処理対象とする.

（3.2）選択された 1 バイトの値が $F0_h$ 以上ならば，その 1 バイトをスキップする（そのブロック B_k をスクランブル処理対象から除外する）.

（4） 手順（3）でスクランブル対象として選択された半バイトを 2 つ用いて 1 バイトを合成し，その 1 バイトを暗号化する.

手順（3）では，**3.1** で基準にした 90_h ではなく $F0_h$ を基準にしている．これは，下位半バイトの選択により $F0_h$ を基準にしても，**3.1** の条件を満たすことができ，かつ $F0_h$ を基準にすることにより，スキップされる 1 バイトの値の範囲を低減することができるためである．次に，スクランブル処理法としてビットシフトを用いる場合について述べる．ビットシフトを用いる場合，選択された半バイトを組み合わせる必要はなく，処理対象として選択された半バイト一つに対し，次のようなビットシフト処理を施す [11].

選択された j 番目の 1 バイトの下位半バイトを

$$X_j = (x_1, x_2, x_3, x_4),\ x_i \in \{0, 1\} \tag{2}$$

とする．この X_j に対して，S_j ビット,

$$S_j \in \{0, 1, 2, 3\} \tag{3}$$

の左回りの巡回シフトを施す．このとき，例えば $S_j = 1$ とすれば，式 (2) の X_j は,

$$\hat{X}_j = (x_2, x_3, x_4, x_1) \tag{4}$$

と変換される.

ビットシフトによるスクランブル処理では，F_h 以外の値は処理後に F_h となることはないことに注意してほしい．したがって，上述の手順（3.1）から（3.2）は省略され，以下の一つの手順にまとめることができる.

（3） 各ブロックごとに選択された 1 バイトに対し，その下位半バイトをスクランブル処理対象として選択する.

また，S_j の値は，乱数列を発生させ，j ごとに可変にすることもできる．固定の場合でも，式 (1) の $\boldsymbol{J}_{p_1}(k)$ が乱数であるので，S_j の値が既知であってもスクランブルの解除は容易ではない．以上の二つの方法は，ともにマーカコードを発生せず，コードストリームのデータ量を変化させない．両者の間には，ビットシフト法は処理が軽く実現が容易であること，暗号化法は暗号化鍵や暗号化法に選択の自由度があり所望の強度をある程度選択できる，という特徴の違いがあると考えている.

3. 2. 2　スクランブル解除法

解除のためには，スクランブル処理で使用した整数乱数発生アルゴリズム $\boldsymbol{J}_{p_1}(k)$，初期値 p_1，L バイトのボディデータ領域（レイヤ，解像度レベル及びコードブロック）及び M の値を知る必要がある．図 3 (b) は復号及びスクランブル解除の手順を示している．まず，符号化側においてスクランブル処理された符号化データを構文解析器に送りヘッダとボディに分解する．次に，スクランブル解除が実行される．以下では，スクランブル処理法として暗号化を選択した場合の解除手順を述べる.

（1） 初期値 p_1 を用いて 1 から M までの整数乱数 $\boldsymbol{J}_{p_1}(k)$ を発生させる．

（2） M バイトの各 B_k において符号化側で選択された $\boldsymbol{J}_{p_1}(k)$ 番目の 1 バイトをそれぞれ特定する．

（3） 各ブロックごとに特定された 1 バイトに対し，次の条件に従ってスクランブル解除の対象となる半バイトをそれぞれ選ぶ．

（3.1）選択された 1 バイトの値が $F0_h$ 未満ならば，その下位半バイトをスクランブル解除対象とする．

（3.2）選択された 1 バイトの値が $F0_h$ 以上ならば，その 1 バイトをスキップする（そのブロック B_k をスクランブル解除対象から除外する）．

（4） 手順（3）で選択された半バイト二つを組み合わせ，1 バイトを合成する．その 1 バイトに対して，暗号鍵を用いて暗号化解除を施す．

スクランブル処理法としてビットシフトを用いた場合には，手順（4）において，$\hat{X}_j$ に S_j ビット分の右回りの巡回シフトを施すことでスクランブル解除を行う．この処理により，$\hat{X}_j$ は X_j となる．また，スクランブル解除の手順（3.1）から（3.2）は省略され，以下の一つの手順にまとめることができる．

（3） 各ブロックごとに特定された 1 バイトに対し，その下位半バイトをスクランブル解除の対象として選択する．

以上の復号手順では，パラメータ M の追加によって，解除のためのパラメータ数が増加し，不正解除がより複雑となる．更に，**5.** で後述するように，パラメータ M の導入は，レイヤ切捨てによる画質改善に対しても有効な手段となる．

復号側において上述したスクランブルの解除を行わず，符号化側でスクランブルされた JPEG2000 ストリームが汎用デコーダに送られた場合，その出力として半開示画像が生成される（図 3 (b) 参照）．

4. 提案法の特長

3. 2 で述べた手順は，**2. 2** で述べた要求条件を満たすことができる．以下で，そのことを具体的に述べる．

（1） コードストリームに対する処理（要求条件 (a) 及び (b)）

提案法は，コードストリームを構文解析器によってヘッダとボディに分解し，ボディにのみスクランブル処理を施す．また，スクランブル処理を施されたコードストリームは，スクランブル処理前のデータ量と同じデータ量となる．したがって，提案法は要求条件 (a) 及び (b) を満たす．このことは，従来法と同様である．

（2） マーカコードの回避（要求条件 (c)）

従来法が 1 バイト単位の処理であったのに対して，提案法は制約付きの半バイト単位の処理である．これにより，提案法は $FF90_h - FFFF_h$ のマーカコードの発生を完全に回避可能である．一方，従来法はマーカコード発生の可能性があり，スクランブル処理後にマーカコードの有無を確認する必要があった．

（3） 制御能力の向上（要求条件 (d) 及び (e)）

スクランブル処理に要する演算量とスクランブルの度合の制御は，レイヤ，解像度レベル及びコードブロックの選択法のみで行ってきた．また，スクランブル強度は，このスクランブル領域の選択と使用するスクランブルアルゴリズムによって決定された．一方，提案法は，更に M という制御パラメータの導入によってこのような制御能力を向上している．

$M = 1$ と固定した場合，提案法のスクランブルに要する演算量とスクランブルの度合に関して提案法と従来法は一致する．M を 1 より大きな値に選ぶほど，スクランブル処理を施されるデータは低減（約 $1/M$ に低減）し，処理に要する演算量を軽減することができる．更に，一般に大きな M の選択は，スクランブルの視覚的程度を弱める．

更に，**5.** で後述するように，M の導入によって，レイヤ切捨てやレベルの切捨て等による画質改善に対しても耐性が向上する．また，パラメータ M の追加と整数乱数発生アルゴリズムの使用（式 (1) 参照）は，解除のためのパラメータ数を増加させ，スクランブルの不正解除をより困難にすると考えられる．

以上のように，パラメータ M の導入によって，要求条件 (d) 及び (e) に関してより制御能力を向上させることができる．この向上の有用性を次の **5.** シミュレーションにおいて確認する．

5. シミュレーション

提案法の効果を確認するため，実際の画像に提案法を適用した結果を示す．シミュレーションには，図 6 に示す Lena の画像（512×512 画素，8 ビットグレースケール）を用いた．JPEG2000 符号化アルゴリズムとしては，JPEG2000 Verification Model 8.6 [18] を用い，Daubechies 9-7 フィルタによる 5 レベル分解，圧縮率 1 (bit/pixel) の符号化を行った．また，スクランブル処理には，ビットシフトを使用した [11]．

図 6 原 画 像
Fig. 6 The original image.

(a) Decoding all layers (b) Decoding only layers 0 and 1

図 7 デコード対象レイヤの選択による視覚的程度の制御（スクランブル位置：レイヤ 2，$M = 1$）
Fig. 7 The example of proposed scramble using JPEG2000 layer structure. (scrambling position: Layer 2, $M = 1$)

5.1 レイヤの選択

まず，スクランブル処理領域としてレイヤを選択した例を示す．レイヤの数及び各レイヤのデータサイズを比較的自由に設定できることから，このレイヤ選択によるスクランブル処理は，画質や演算量の制御が容易であり，有用な方法である．この例では，符号化時に 0.02 (bit/pixel) 刻みに 50 層のレイヤ（レイヤ 0 からレイヤ 49）を形成した．図 7 (a) は，レイヤ 2（最上位レイヤ 0）をスクランブル対象領域として選択した例である．$M = 1$ の選択により，従来法と同様，対象領域全体（レイヤ 2）のボディデータがスクランブルされ，全レイヤ，すなわち符号化データ全体がデコードされている．画質は従来法と同等であるが，マーカコードの発生を完全に回避できる．より上位のレイヤを選択すれば，スクランブルの度合を強めることもできる．

図 7 (b) は，スクランブル処理を施したレイヤ 2 以下をデコードせず，レイヤ 2 より上位のレイヤ 0 及び 1 のみをデコードした例である．同図 (a) に比べ画質が改善されているのがわかる．これは，レイヤ 2 以降がノイズとして振る舞い，画質を劣化させていたのに対して，下位レイヤの切捨処理によって，そのノイズが除去されたためである．このことがレイヤを処理領域とした場合の大きな欠点である．すなわち，スクランブル処理されていない上位レイヤのみをデコードし，スクランブル処理されたレイヤ以下を切り捨てることにより，画質を改善でき，スクランブルの効果を容易に低減されてしまう．このことの回避のためには，従来法ではレイヤではなく，自由度の少ない解像度レベルをスクランブル処理領域とする必要があった．図 8 (a) は，最上位レイヤであるレイヤ 0 をスクランブル対象領域として選択した例である．レイヤ 0 の選択は，下位レイヤ切捨てによる画質改善を回避できる．しかし，従来法 ($M = 1$) では，レイヤ 0 の選択は非開示画像を生成してしまう．一方，提案法では，同図 (b) に示すように，レイヤ 0 を選択しても適当な M の選択によって，半開示画像を生成することが可能となる．

(a) $M = 1$ (b) $M = 128$

図 8 レイヤ及び M の値の選択による視覚的程度の制御（スクランブル位置：レイヤ 0）
Fig. 8 The example of proposed scramble using SNR scalability function and parameter M. (scrambling position: Layer 0)

5.2 コードブロックの選択

次にコードブロックをスクランブル処理領域に選択した例を示す．この例では，コードブロックサイズを 16×16 とし，解像度レベル 3，4 及び 5 のコードブロックのうち，顔の領域に対応するものを選択し，スクランブル処理を施した．図 9 (a) では $M = 1$ を，同図 (b) では $M = 32$ をそれぞれ選択している．コードブロックの選択により，ある特定の領域のみの半開示が可能であり，かつ，M の値の選択により視覚的程度の制御が可能であることが確認できる．当然，マー

(a) $M = 1$　　(b) $M = 32$

図 9　スクランブルする画像領域の制御（中央部を選択）
Fig. 9　The example of region-scramble using proposed method. (The scrambled region is the center of the image.)

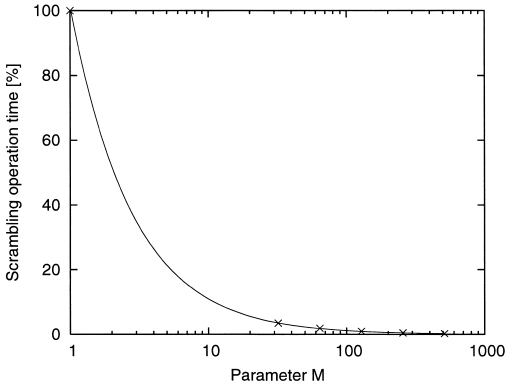


図 10　パラメータ M によるスクランブル処理時間の制御例
Fig. 10　The processing time of proposed scrambling which is controlled by parameter M.

カコードの発生の可能性もない．

5.3　処理時間の制御

動画像のように実時間処理が強く求められる応用においても，種々の画像サイズの下でほぼ一定の処理時間を保証することが可能となる．提案法によって導入したパラメータ M は，画像の制御のみならずスクランブルに要する処理時間の制御にも有効である．

図 10 はスクランブル手法にビットシフトを用いた場合に，それぞれの M の値のおけるスクランブル処理時間を実測し，その結果を示したものである．横軸は M の値，縦軸は $M = 1$ のときのスクランブル処理時間を 100%とした場合の各 M の値におけるスクランブル処理時間である．同図より，各 M の値において，スクランブル処理に要する時間は $M = 1$ のときの約 $1/M$ となり，M の値によりスクランブル処理時間の制御が可能であることがわかる．以上のことから，提案法は，マーカコードを発生することなくスクランブル領域，視覚的程度及び演算量の制御を行うことができ，用途に応じた適切なスクランブル処理が可能であることがわかる．すなわち，提案法は，M の値を利用することにより，演算量が画像サイズに依存するという問題を回避している．図 7，図 8 及び図 9 に示したいずれの場合も，スクランブル処理において使用したパラメータ，初期値及び M の値を用いれば，スクランブルを解除でき，スクランブル処理を行わなかったときに生成される画像を得ることができる．

6. むすび

本論文では，JPEG2000 符号化画像において，マーカコードの発生を回避し正常な再生が可能であるとともに画質，演算量及びスクランブル強度の制御能力を向上した情報半開示方式を提案した．ボディデータに対し制約付き半バイト単位の処理を実行したことと，パラメータ M の導入によって上述のことが可能となった．また，シミュレーションにより半開示の視覚的程度や，スクランブル処理を行う画像領域が選択可能であることを示し，提案法の有効性を確認した．

謝辞　本研究を進めるにあたり有益な助言を頂いたソニー株式会社プラットフォームテクノロジーセンターメディアテクノロジー部門福原隆浩氏，木村青司氏，安藤勝俊氏に感謝する．

文　献

[1] 松井甲子雄，電子透かしの基礎—マルチメディアのニュープロテクト技術，森北出版，1998.
[2] G.C. Langelaar, I. Setyawan, and R.L. Lagendijk, “Watermarking digital image and video data,” IEEE Signal Process. Mag., vol.17, no.5, pp.20–46, Sept. 2000.
[3] 木下宏揚，塩入律雄，酒井善則，“DCT 符号化に適した画像暗号化方式の提案，” 信学論 (D-I), vol.J75-D-I, no.5, pp.314–321, May 1992.
[4] 藤井　寛，山中康史，“ディジタル画像情報流通支援のためのスクランブル方式，” 情処学論，vol.38, no.10, pp.1945–1955, Oct. 1997.
[5] 藤井　寛，阿部剛仁，西原祐一，串間和彦，“情報半開示方式，” NTT R&D, vol.47, no.6, pp.705–710, June 1998.
[6] R. Grosbois, P.Gerbelot, and T. Ebrahimi, “Authentication and access control in the JPEG2000 compressed domain,” Proc. SPIE 46th Annual Meeting, Applications of Digital Image Processing XXIV, vol.4472, pp.95–104, San Diego, July 29th–Aug. 3rd, 2001.
[7] 安藤勝俊，渡邊　修，貴家仁志，“JPEG2000 に基づく

静止画像の半開示方式," 信学技報, DSP2000-106, Oct. 2000.
[8] K. Ando, O. Watanabe, and H. Kiya, "Partial-scrambling of still images based on JPEG2000," International Conference on Information, Communications, and Signal Processing, 2F2.5, CD-ROM, Singapore, Oct. 2001.
[9] 安藤勝俊, 渡邊 修, 貴家仁志, "JPEG2000 符号化画像の情報半開示法," 信学論 (D-II), vol.J85-D-II, no.2, pp.282–290, Feb. 2002.
[10] 安藤勝俊, 貴家仁志, "レイヤ構造を利用した JPEG2000 符号化画像の暗号化法," 信学論 (A), vol.J85-A, no.10, pp.1091–1099, Oct. 2002.
[11] 貴家仁志, 今泉祥子, "マーカコードの発生を考慮した JPEG2000 符号化画像の情報半開示法," 信学技報, IE2002-27, June 2002.
[12] T. Fukuhara, K. Ando, O. Watanabe, and H. Kiya, "Partial-scrambling of JPEG2000 images for security applications," ISO/IEC JTC 1/SC29/WG1, N2430.
[13] T. Fukuhara, K. Ando, and H. Kiya, "Proposal of conditional access and signaling of security parameter for JPSEC," ISO/IEC JTC 1/SC29/WG1, N2721.
[14] ISO/IEC IS 15444-1, "Information technology — JPEG2000 image coding system — Part 1: Core coding system," 2000.
[15] D. Taubman and M. Marcellin, JPEG2000 Image Compression Fundamentals, Standard and Practice, Kluwer Academic Publishers, Jan. 2002.
[16] D. Taubman, "High performance scalable image compression with EBCOT," IEEE Trans. Image Process., vol.9, no.7, pp.1158–1170, July 2000.
[17] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," Fast Software Encryption, Cambridge Security Workshop Proceedings (Dec. 1993), pp.191–204, Springer-Verlag, 1994.
[18] ISO/IEC JTC 1/SC29/WG1 WG1 N1894, "JPEG2000 Verification Model 8.6 Software," 2000.

（平成 15 年 2 月 24 日受付，6 月 17 日再受付）

貴家 仁志 （正員）

1980 長岡技科大・工・電気電子システム卒．1982 同大大学院修士課程了．同年東京都立大工学部電気工学科助手．2000 同大大学院電気工学専攻教授．工博．1995〜1996 シドニー大（オーストラリア）客員研究員．マルチレート信号処理，画像符号化及びメディアセキュア技術に関する研究に従事．IEEE 論文誌 Signal Processing 編集委員 (1998〜2000)．本会和文論文誌 A 編集委員 (1998〜2002)．著書「高速フーリエ変換とその応用」，「ディジタル信号処理」，「マルチレート信号処理」．電子画像学会，映像情報メディア学会各会員，IEEE シニア会員．

今泉 祥子 （学生員）

2002 東京都立大・工・電気卒．2003 同大大学院工学研究科修士課程中退．同年文部科学省入省．現在，科学技術・学術政策局に所属．

渡邊 修 （学生員）

1999 東京都立大・工・電子情報卒．2001 同大大学院工学研究科修士課程了．現在，同大大学院工学研究科博士課程在学中．主として，画像符号化の研究に従事．IEEE 会員．