

フラクタル符号化特徴量を用いた類似画像検索 およびオブジェクト検出手法の検討

鶴見 智*

(2019年1月7日受理)

1. はじめに

近年第4次産業革命と呼ばれる社会変革の中でとりわけビッグデータの利活用が重要視されている。急速なIoTの進展やネットワークの高度化に伴い、スマートフォン等からは位置情報、行動履歴が、インターネットやテレビからは視聴・消費行動の情報が、防犯カメラや車載搭載カメラからは画像データが非構造的に日々蓄積されている。これらビッグデータを分析・活用するための手段としてAI（人工知能）が注目されている[1]。

画像データに注目したときにビッグデータ規模の画像データから類似画像検索、物体検出、物体認識、物体追跡を実用的な時間と精度で実現することは極めて難しい。その要因のひとつは対象となるデータの規模にある。そのため画像データは符号化したままで処理できることが望ましい。米山らはMPEG動画データから直接的に移動体物体を高速に検出する手法を提案した[2]。しかしながらこの手法は動きベクトル情報を用いているため、動きベクトルが失われるフレームでは検出が困難となる。岩崎らは「特徴領域画像」をあらかじめ生成することでこの問題に対処する手法を提案した[3]が、前処理を必要とするため符号データそのままで処理とはいえない。

本研究ではMPEGとは根本的に手法の異なるフラクタル符号化[4]に着目する。フラクタル符号化は画像がもつ自己相似性を利用して自身の近似画像を生成する変換符号化法であり、高圧縮率と高速復号を特徴とする。生成される符号は画像内の自己相似情報そのものであり、画像の構造的な特徴を含んでいる。符号化の過程で得られるドメイン-レンジ・ブロック間の誤差（コラージュ誤差と呼ばれる）ヒストグラムは画像の持つ自己相似性の度合いを特徴付け、画像により異なる[5]。一方類似の画像は似た分布となるため、画像の検索やオブジェクトの検出に用いることができる。

本研究ではフラクタル符号に含まれるコラージュ誤差ヒストグラムとブロック分割構造を特徴量として用いる新しい類似画像検索および動画中のオブジェクト検出手法を提案する。符号に含まれる特徴量を復号することなく直接利用するため、低容量の状態でも高速に処理を行うことが可能である。

論文の構成は以下の通りである。

第2章でフラクタル符号化の原理を述べる。第3章ではフラクタル符号を用いた類似画像検索の原理を述べ、評価実験を行った結果を示す。第4章ではフラクタル符号により生成されるブロック分割特徴量について述べ、動画中のオブジェクト検出の原理と実験結果を示す。最後にMPEGを用いた既存手法との比較をし提案手法の有効性について考察する。

2. フラクタル符号化[4]

2.1 フラクタル符号化

フラクタル符号化は画像の持つ自己相似性を利用して自身の近似画像を生成する画像圧縮手法である。図1に示すように対象画像から互いに重複しない $N \times N$ の大きさのレンジブロック R_i と $2N \times 2N$ の大きさのドメインブロック D_i を取り出す。ドメインブロック D_i に対して縮小アフィン変換

$$\begin{pmatrix} D'_{ix} \\ D'_{iy} \\ D'_{iz} \end{pmatrix} = \begin{pmatrix} 0.5 \cos \theta & 0.5 \sin \theta & 0 \\ -0.5 \sin \theta & 0.5 \cos \theta & 0 \\ 0 & 0 & s \end{pmatrix} \begin{pmatrix} D_{ix} \\ D_{iy} \\ D_{iz} \end{pmatrix} + \begin{pmatrix} e \\ f \\ o \end{pmatrix} \quad (1)$$

を施し変換ブロック D'_i を生成する。ここで (D_{ix}, D_{iy}) および (D'_{ix}, D'_{iy}) はそれぞれ縮小アフィン変換前後のドメインブロックの座標値を表す。また D_{iz} と D'_{iz} はそれぞれ D_i と D'_i の輝度値である。パラメータ θ, s, e, f, o は回転移動や輝度の一次変換を表す。

*電子情報工学科

次に得られたドメインブロック D_i' とレンジブロック R_i との二乗和誤差

$$\Delta^2 = \sum_i^N \sum_j^N (s \cdot d_{ij} + o - r_{ij})^2 \quad (2)$$

が最小となるドメインブロックとその変換の組み合わせを画像全領域から探索する。ただし、元画像中の位置を (i, j) としたとき、 d_{ij} はドメインブロック中の (i, j) における画素値を表し、 r_{ij} はレンジブロック中の (i, j) における画素値を表す。 s, o はそれぞれアフィン変換における輝度スケーリング、輝度シフトを表す。この探索をすべてのレンジブロックに対して行う。

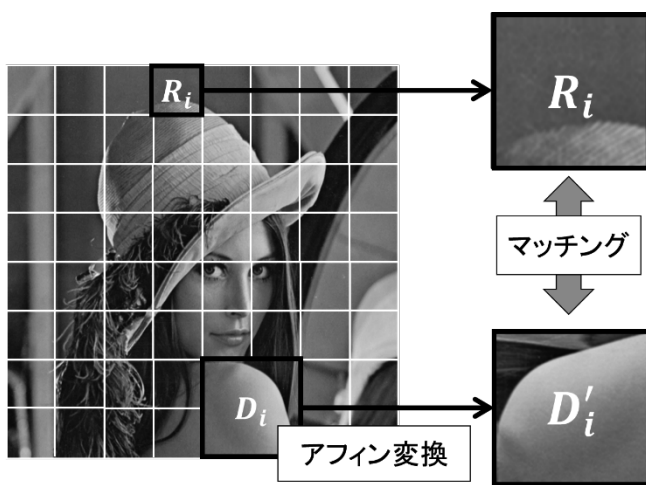


図1 フラクタル符号化

フラクタル符号化は、以上のパラメータをバイナリ一符号とするため符号化容量は小さく、復号は任意の画像を初期値として各レンジに対するアフィン変換を数回施すことで得られるため極めて高速である。一方レンジブロックとドメインブロックに対し、存在する全ての組み合わせについて探索を行うため、貪欲アルゴリズムでは符号化時間は JPEG・MPEG に比べかなりかかる。我々はこの問題に対して GPU・MPI での実装と粒子群最適化法 (PSO) アルゴリズムを適用して実用レベルの高速化を達成している [6, 7]。

2.2 四分木分割アルゴリズム

2.1 のアルゴリズムではレンジブロックのサイズは固定されているためこのままでは画質は実用レベルには達しない。画質向上のために四分木分割アルゴリズムが使われる。式(2)の最小値に閾値を設ける。閾値は任意に設定可能であり、これを満たさない場合は当該レンジブロックのサイズを4分割(再分割と呼ぶ)し、その全てに対して再び探索を行う。再分割処理の過程を図2に

示す。再分割が行われるごとにレンジブロック、ドメインブロックのサイズは半分になる。閾値を満たすか、指定したブロックサイズになるまで反復する。

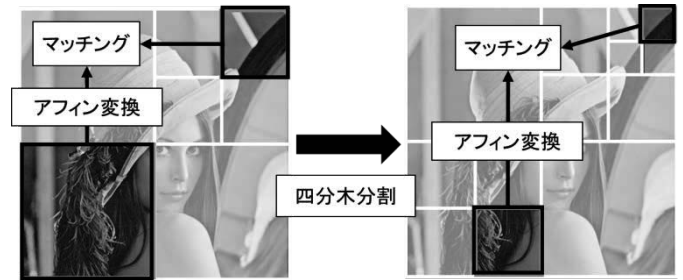


図2 四分木分割アルゴリズム

実際に四分木分割アルゴリズムによって生成されたブロック画像を図3に示す。エッジ部分については小さく分割され、輪郭がわかるようになっている。また、それ以外の部分についてはエッジ部と同様、もしくは大きいブロックサイズで分割されている。

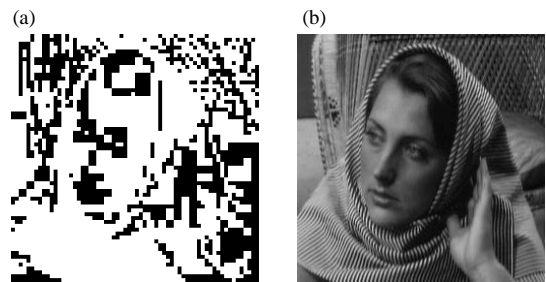


図3 四分木分割ブロック画像(a)と原画像(b)

2.3 コラージュ誤差ヒストグラム

コラージュ誤差は式(2)を用いて(3)式で定義される。

$$CollageError = \frac{1}{N} \|\Delta\| \quad (3)$$

ただし $\|\cdot\|$ は L^2 ノルムを表す。コラージュ誤差は縮小アフィン変換後のドメインブロックとレンジブロック間の画素値の二乗平均平方根誤差を意味する。コラージュ誤差を用いると画像ごとのコラージュ誤差ヒストグラムを求めることができる。図4にコラージュ誤差ヒストグラムの例を示す。

コラージュ誤差ヒストグラムは、相似領域間における輝度分布の差異を表し、その分布は画像のもつ背景やエッジの量に依存する。そのため、コラージュ誤差ヒストグラムを比較することで、類似画像の検索が可能である。また、複数のオブジェクト(人、車等)があるような画像では、オブジェクトが検出できればオブジェクトごと

のカラーズ誤差ヒストグラムを用いてオブジェクトの識別が可能である。

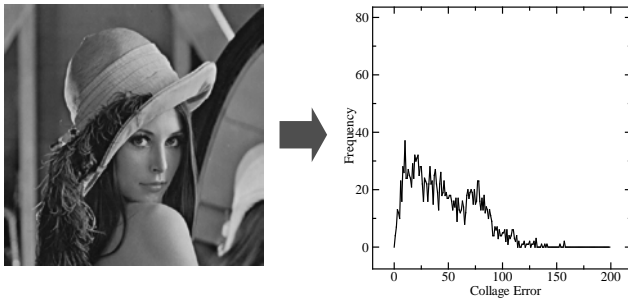


図4 カラーズ誤差ヒストグラム

3. フラクタル符号化を用いた類似画像検索

3.1 関連研究

関連研究として、横山らの自己相似性を持つブロックの位置関係を代表ベクトルで表現し画像同士のベクトル分布を比較することで検索を行ったものがある[8]。しかしカラーズ誤差ヒストグラムそのものを比較に用いている研究はない。

3.2 類似画像検索の原理

カラーズ誤差ヒストグラムより6つの統計的指標(平均 μ , 分散 σ^2 , 歪度 S , 尖度 K , エネルギー $EntY$, エントロピー EPY)をそれぞれ抽出し、ヒストグラムの形状を特徴付ける。これにより各ヒストグラムを6次元ベクトルによって近似的に表現することができる。類似度をベクトルの距離で定義すると、形状が類似している2つのヒストグラムのベクトル間距離は小さい値をとる。なお、これらの指標は全体が1.0になるようにヒストグラムを正規化してから求めている。

3.3 評価実験

画像データベース(SIMPLIcity project [9])を用いて類似画像検索の評価実験を行った。このデータベースには計1000枚の画像が10カテゴリに100枚ずつ分類されており、同じカテゴリに含まれる画像は互いに意味的に類似しているのでこの実験に適している。実験は各カテゴリから1枚ずつ任意に選択した画像をクエリ画像とし、適合画像の検索数に応じて行った。適合画像はクエリ画像と同じカテゴリに含まれている画像とした。性能評価には検索システムの一般的な性能評価指標である適合率

precision

$$= \frac{|\{\text{データベース適合画像}\} \cap \{\text{検索された画像}\}|}{|\{\text{検索された画像}\}|}$$

(4)

を用いた。ここで{・}は画像の集合を意味する。適合率は検索画像数に対する正しく検索された画像の割合を意味する。表1に実験結果を示す。

表1 類似画像検索の適合率

カテゴリ	適合率 [%]
Africa people and village	85
Beach	40
Buildings	40
Buses	50
Dinosaurs	100
Elephants	15
Flowers	75
Horses	100
Mountains and glaciers	10
Food	100
平均	61.5

実験結果より“Dinosaurs”, “Horses”, “Food”に含まれる画像をクエリ画像とした場合適合率は100%となることがわかった。一方“Elephants”や“Mountains and glaciers”では低い値となっている。これは画像によっては見た目には類似しているが意味的類似度は低い画像が他カテゴリにも多く存在し、異なるカテゴリ間における各ヒストグラムの類似度が高いことが理由と考えられる。すなわちカラーズ誤差ヒストグラムが似ていても意味的に差異があるセマンティックギャップが生じていると考えられる。図5に異なるカテゴリに属する類似ヒストグラムの例を示す。

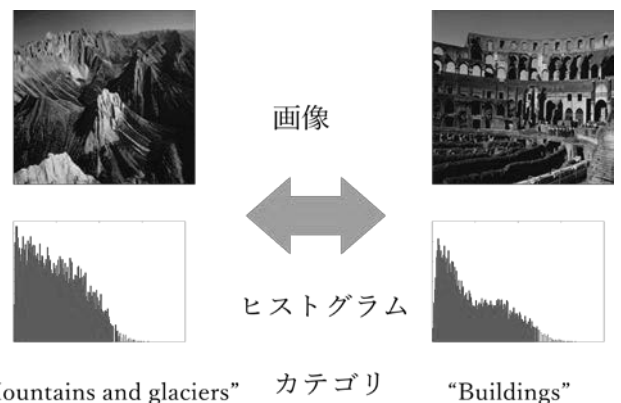


図5 異なるカテゴリに属する類似ヒストグラムの例

4. フラクタル符号特徴量

4.1 フラクタル符号の構造的特徴

2.2で述べたように四分木分割フラクタル符号化で

は符号化過程でブロック分割情報が生成される。背景、壁といったエッジの少ない部分では分割数が少なく大きなブロックのままであり、オブジェクト周りのようにエッジが多い部分では分割数が多くなり小さなブロックの集合となる。このブロック分割情報を用いて背景とオブジェクトを分離することが可能となる。図6にブロック分割情報の例を示す。

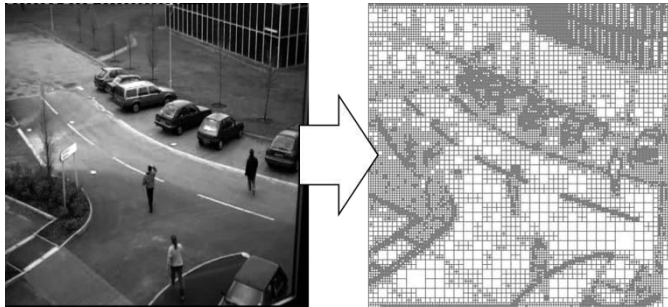


図6 フラクタル符号で得られるブロック分割情報

4.2 背景差分によるオブジェクト検出

動画はいくつかの連続するフレームからなり通常30～60fpsある。各フレームをフラクタル符号化することで動画を符号化できる。ここではすでに各フレームはフラクタル符号化されているとする。ブロック分割情報を用いてオブジェクト領域と背景を分離するため、動的に背景ブロックを生成する。生成される可能性のある全ブロックについて、現在のフレーム以前のいくつかのフレームの間で生成された回数をカウントする。生成される回数が多いブロックはフレーム間で動きが少ない部分と考え、回数に閾値を設定してその閾値を超えるブロックを背景ブロックとして扱う。

生成された背景ブロックと現在のフレームにおけるフラクタルブロックとの差分をとることで、現在のフレームから背景領域を取り除くことができる。背景領域を取り除き、ある大きさのブロックの隣接するブロック同士を合わせて一つのクラスタとする。図7に背景差分をとってオブジェクトを抽出した例を示す。

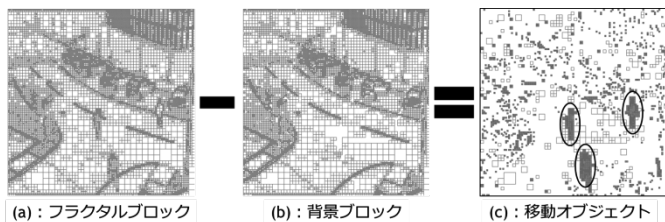


図7 背景差分によるオブジェクト抽出

あるフレームで抽出された領域におけるコラージュ誤差ヒストグラムと、前フレームで抽出された領域にお

けるコラージュ誤差ヒストグラムとの類似度を計算し、同一オブジェクトとみられる抽出領域のヒストグラムの類似度が他の抽出領域におけるヒストグラムの類似度と比較して最良となっていれば、フラクタル符号から直接的に移動オブジェクトの追跡が行うことができる。

4.3 実験

提案手法によるオブジェクトの検出の有効性を検証するため、動画中の移動オブジェクトの追跡実験を行った。実験ではPETS'2000 [10]の動画（ただし768×576画素のものを512×512画素にリサイズした）を対象として、0.5秒間隔であらかじめフラクタル符号化したものをフラクタル符号ストリームとして用いた。この動画は定点映像であり、単体または複数の歩行者と自動車が移動する様子が記録されている。実験環境は表2のとおりである。

表2：実験環境

OS	Ubuntu 14.04 LTS 64bit
CPU	Intel Core i7-4700MQ 2.40GHz
メモリ	16GB

図8.1、図8.2に自動車と歩行者の検出と追跡の様子を示す。オブジェクトの追跡が正確に行われていることがわかる。

次に表3.1、表3.2に追跡の精度と処理速度の結果を示す。ただし、歩行者と自動車の判別は手動で行った。

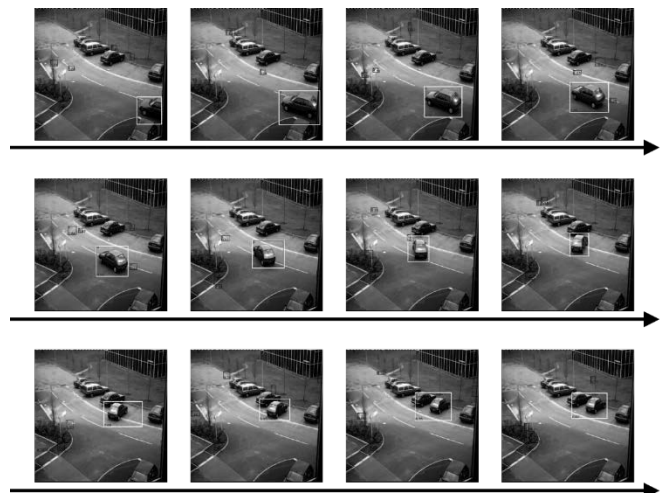


図8.1 自動車の追跡実験

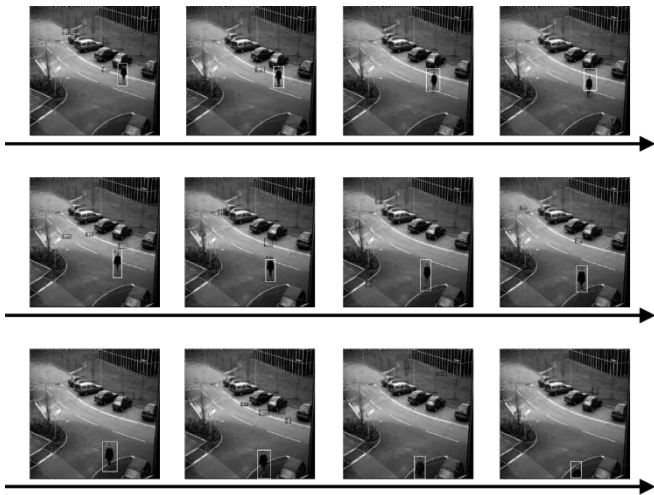


図 8.2 歩行者の追跡実験

表 3.1 : 追跡実験の結果 (精度)

	成功[frame]	失敗[frame]	精度[%]
歩行者	117	23	83.6
自動車	86	6	93.5
計	203	29	87.5

表 3.2 : 追跡実験の結果 (処理速度)

	検出	追跡	計
処理時間[ms]	0.9444	0.4753	1.4197

実験結果より歩行者で約 84%、自動車で約 94%の精度で検出・追跡が行えていることがわかる。歩行者において精度が落ちているのは背景差分による検出精度が自動車より歩行者の方が低いことが原因と考えられる。検出精度を改善すれば十分実用レベルの追跡精度を達成できると考えられる。

また処理速度は検出と追跡の総計で約 1.4ms と高速であり、フレームレートでは平均 700fps の速度で追跡処理を実行できている。Aggarwal らの MPEG を用いた手法 [11]での処理速度は平均で約 90fps 程度であったことから、本手法は MPEG 手法より処理速度で優位であると考えられる。

5. まとめ

フラクタル符号に含まれるカラーズ誤差ヒストグラムとブロック分割構造を特徴量として用いる新しい類似画像検索および動画中のオブジェクト検出手法を提案し、実験を行った。

類似画像検索実験ではヒストグラムの類似性を用いることで平均約 62%の適合率を達成できたが、カテゴリによって 100%の画像がある一方 15%の画像もあった。

一般にはこの手法では意味的に差異がある、いわゆるセマンティックギャップが生じてしまっている。セマンティックギャップの解消をするには、ヒストグラム以外の情報を付加する方法とともに、対話型遺伝的アルゴリズム[12]の導入も有効であると考えられる。

また動画中のオブジェクト (歩行者と自動車) 検出と追跡実験では約 88%の検出精度を達成できた。特に処理速度では約 1.4m/s を達成し、フレームレートの比較では MPEG 手法より高速であることが示された。

今後の課題としては、動画中で類似のオブジェクトが交差する場合やさまざまな環境条件でのロバスト性の検証がある。最終的な目標は高速かつ高精度で動画中のオブジェクトの検出・認識を実現することである。

6. 謝辞

本研究について有益な議論をしていただいた栃原直哉、石原慎の両氏に深く感謝する。

参考文献

- 1) 人工知能学会誌 Vol.28, No.1, 特集: 「ビッグデータと AI」 pp.82-137, 2013.
- 2) 米山暁夫, 中島康之, 柳原広昌, 菅野 勝, “MPEG ビデオストリームからの移動物体の検出”, 電子情報通信学会論文誌. Vol.J81-D-II, No.8, pp.1776-1786, 1998.
- 3) 岩崎敏紀, 横山貴紀, 渡辺俊典, 古賀久志, “MPEG ビデオデータの動きベクトルを用いた圧縮領域における移動物体の検出と追跡”, Vol.J91-D No.6 pp.1592-1603, 2008.
- 4) Y. Fisher (ed.), Fractal Image Compression :Theory and Application, Springer-Verlag, 1994.
- 5) S.K.Alexander, E.R.Vrscay, S.Tsurumi, “A simple, yet general, model for the affine self-similarity of images”, Image Analysis and Recognition, Springer Berlin/Heidelberg, pp192-203, 2008.
- 6) 鶴見智, “フラクタル符号化における並列計算の検討 -GPU, MPI による実装と評価-”, 群馬高専レビュー, No.32, pp.49-53, 2013.
- 7) 鶴見智, 石橋諒馬, “粒子群最適化法 (PSO) フラクタル符号化の実装と GPU による高速化”, 群馬高専レビュー, No.34, pp.65-71, 2015.
- 8) 横山貴紀, 菅原研, 渡辺俊典, “フラクタル符号に基づく圧縮領域における類似画像検索手法”, 情報処理学会論文誌, Vol.45, No.SIG 4(TOD 21), pp.11-22, Mar. 2004.
- 9) J.Z. Wang, J. Li, and G. Wiederhold, “SIMPLiCity: Semantic sensitive integrated matching for picture libraries”, IEEE Trans. Pattern Anal. Mach. Intell., 23(9):947-963, Sep. 2001.
- 10) PETS'2000 : Performance Evaluation of Tracking and Surveillance, 2000, (<http://ftp.pets.reading.ac.uk>)
- 11) Ashwani Aggarwal, Susmit Biswas, Sandeep Singh, Shamik Sural, and A.K. Majumdar, “Object Tracking Using Background Subtraction and Motion Estimation in MPEG Videos”, ACCV2006, LNCS3852, pp.121-130, 2006.
- 12) C.-C. Lai and Y.-C. Chen, “A User-Oriented Image Retrieval System Based on Interactive Genetic Algorithm”, EEE Trans. Instrum. Meas., 60(10):3318-3325, Oct. 2011.

Study on Image Retrieval System and Object Detection Method Using Fractal Coding Feature

Satoshi TSURUMI

Fractal image coding is a block-based scheme that exploits the self-similarity hiding with an image. Fractal codes are quantitative measurements of the self-similarity of the image, and collage error distribution of block characterizes the degree of self-similarity in it. Furthermore, fractal codes can be used to obtain a practical image indexing system because of its compactness and stability. The most important reason using fractal codes is able to deal with the images in compressed form. Thus fractal indexing is suitable for use with large database. In this study, we propose a new image retrieval system and object detection method based on fractal coding features that are collage error distribution and block partition structure in fractal codes. Experimental results show that the proposed method achieves a high precision tracking which is faster than MPEG method.