

PSO フラクタル符号化の GPU による高速化

石橋 諒馬[†] 鶴見 智[‡]群馬工業高等専門学校専攻科生産システム工学専攻[†]群馬工業高等専門学校電子情報工学科[‡]

1. はじめに

フラクタル符号化は、画像の自己相似性を用いて近似画像を生成する画像データ圧縮方式のひとつである。この手法は高圧縮率を誇る一方で、符号化時間などのパフォーマンスで従来の圧縮方式に劣るため、普及に至っていない現状がある。

本研究では、フラクタル符号化の一般的手法である四分木分割アルゴリズムに対し、最適化手法の一種である粒子群最適化法 (Particle Swarm Optimization: PSO) を用いた高速化を提案する。複数の画像を用いた実験により汎用パラメータを決定し、評価実験を行うことで本手法の高速化に対する有用性を考察する。また、Graphics Processing Unit (GPU) を用いた PSO の並列化についても検討する。

2. 四分木分割によるフラクタル符号化

フラクタル符号化は、画像に存在する自己相似性を利用して、自身の近似画像を生成することにより行われる [1]。

原画像から互いに重複しない $N \times N$ 画素のレンジブロック及び $2N \times 2N$ 画素のドメインブロックを取り出し、ドメインブロックに縮小及び回転・反転の変換を行い、レンジブロックとの 2 乗和誤差が最小となる組み合わせを比較し決定する。2 乗和誤差は以下の式で表される。

$$\sum_i \sum_j (s \cdot d_{ij} + o - r_{ij})^2 \quad (1)$$

座標を (i, j) とし、 d_{ij} は (i, j) でのドメインブロックの画素値、 r_{ij} は (i, j) でのレンジブロックの画素値である。 s は輝度スケール、 o は輝度シフトを表わしている。

四分木分割アルゴリズムは 2 乗和誤差に閾値を設け、満たさない場合はブロックを再分割し、比較を行う方法である [2]。変換・比較処理を行い (図 1-a)、閾値を満たさない場合には、ドメイ

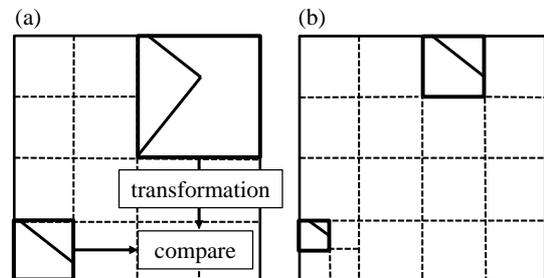


図 1 再分割処理

ン・レンジの両ブロックを半分のサイズに分割し、同様の処理を行う (図 1-b)。一般に、ブロックサイズが小さくなるほど近似精度は向上するので、再分割処理を行うことにより、高画質な符号化が期待できる。

3. 粒子群最適化法 (PSO) の実装

PSO は、粒子群の行動を模倣して最適化問題の解探索を行うアルゴリズムである [3]。問題空間上に存在する複数の粒子は、個々が位置ベクトルと速度ベクトル、その位置における価値 (最適化問題の解) を保持しており、粒子はこの情報を共有しながら最適解に近づくよう行動する。現在の位置ベクトル p 及び速度ベクトル v は、行動後を p' 、 v' として次式で示される。

$$v' = \omega \cdot v + r_1 (pbest - p) + r_2 (gbest - p) \quad (2)$$

$$p' = p + v' \quad (3)$$

ここで、 $pbest$ は現在の行動で最良の価値を持った位置を表し、 $gbest$ は繰り返した全行動中で最良の価値を持った位置を表す。 ω (慣性重み) は現在の速度を重視する割合であり、1.0 以下の任意数を設定可能である。 r は 1.0 以下の乱数であり、これは、速度が $pbest$ や $gbest$ に依存する割合となっている。

フラクタル符号化は、レンジ・ドメインブロック間誤差が最小となる組み合わせを見つける最適化問題と考えることができるため、PSO を用いて最適化を行うことが可能である。原画像上に複数の粒子を生成し、ドメインブロックの左下座標を位置ベクトル (p_x, p_y) として割り当てる。それぞれの粒子はランダムな位置ベクトル、速度ベクトルを持つように初期化され、式 (2)、

Fast PSO-Fractal Coding Using GPU

[†]Ryoma IshibashiAdvanced Production Systems Engineering Course,
Gunma National College of Technology[‡]Satoshi TsurumiDepartment of Information and Computer Engineering,
Gunma National College of Technology

(3)に従って行動する。任意の行動回数これを繰り返し、2乗和誤差が閾値を満たさなければ通常の四分木分割アルゴリズムと同様に再分割を行う。指定されたサイズまで分割されるか、閾値を満たせばその変換係数を用いて符号化を行う。また、一般的に四分木分割ではドメインサイズが小さいほど2乗和誤差が少なくなるという傾向がある。そのため提案法では、再分割されるごとに粒子数や行動回数を制御し、効率化を図っている。

4. 実験

4.1 予備実験

提案法には、行動回数やブロックサイズなどを制御するパラメータが存在する。そのため、予備実験として汎用的なパラメータの決定を行った。選定の際はSIDBA標準画像(512×512, 8bpp グレースケール)を用いた。人物画像としてLena, 背景画像としてBoat, エッジ部分の多い画像としてMandrillを使用している。2乗和誤差の閾値は2.0と設定した。

実験環境を表1に、パラメータの選定結果を表2に示す。制限速度は粒子の行動を制限する閾値である。粒子数は分割数の増加に従って5ずつ増加させ、行動回数は2の累乗ずつ増加させる。

4.2 評価実験

4.1において決定したパラメータを用いて評価実験を行った。6種類のグレースケール画像を用いて提案法の性能を評価した。実験環境及び条件は4.1と同様である。

表3に実験結果を示す。PSOを用いた提案法では、通常の四分木分割アルゴリズムと比較してPSNRが1.0[dB]程度劣る結果となったが、一方で符号化時間は平均71%高速化された。粒子数・行動回数を増加させることによりPSNRを通常法に近づけることも可能だが、高速化の観点からこのパラメータとしている。

5. GPUによるPSO並列化

PSOは複数の粒子によって解を探索するといった考え方であり、その特徴から並列計算に向いたアルゴリズムである。そのため、並列処理に特化したGPUを用いて符号化することでさらなる高速化が見込める。

GPUでは、並列処理をスレッド・ブロック・グリッドという単位で分割している。実行の際には、スレッドが処理を実行することになる。PSOでは、行動空間を分割してスレッドに割り当てる行動空間分割と、粒子それぞれをスレッドに割り当てる粒子処理分割が考えられる。PSO

表1 実験環境

OS	Windows7 Professional 64bit
CPU	Intel Core i5-2400 3.10[GHz]
Memory	16[GB]
Compiler	gcc ver.4.3.3

表2 パラメータ

分割数	粒子数	慣性重み	制限速度	行動回数
3	10	1.0	32	2
4	15			4
5	20			8
6	25			16
7	30			32

表3 実験結果

画像	通常法		提案法	
	PSNR[dB]	時間[s]	PSNR[dB]	時間[s]
Lena	36.9	50.0	36.0	12.5
Mandrill	25.1	59.0	24.1	20.6
Boat	35.1	42.0	34.1	12.0
Peppers	36.2	56.0	35.0	15.7
Goldhill	34.9	54.0	34.0	17.3
Barbara	30.0	51.0	28.7	13.7

では粒子の位置ベクトルは随時変更されるため、行動空間分割は非効率的である。従って、粒子処理分割を行うことで並列化することとした。GPUによる実装及び評価結果は発表当日に示す。

6. まとめ

フラクタル符号化の高速化のために、最適化手法であるPSOを四分木分割アルゴリズムに適用する手法を提案した。ドメインブロックにPSOの粒子をそれぞれ割り当て、粒子数や行動回数などのパラメータを制御することで実現した。これにより、画質の劣化を1.0[dB]程度に抑えながら符号化時間を大幅に高速化できることを確認し、PSOの有用性を示した。

また、GPUを用いてPSOにおける粒子行動を並列化することによる高速化についても検討、実装した。

参考文献

- [1] M.F. Barnsley, Fractal Everywhere, Academic Press, 1989.
- [2] Y. Fisher (ed.), Fractal Image Compression :Theory and Application, Springer-Verlag, 1994.
- [3] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization," Proc. IEEE International Conference on Neural Networks, vol.4, pp.1942-1948, Perth, Australia, 1995.