

GAを用いた動画像符号化のための領域分割

高木 幸一 小池 淳 松本 修一

(株)KDD 研究所 画像通信グループ

〒 356-8502 埼玉県上福岡市大原 2-1-15

E-mail : ko-takagi@kdd.co.jp

あらかし 動画像の符号化のための遺伝的アルゴリズム (Genetic Algorithm:GA) を用いた領域分割を行う方法を提案する。動き補償予測符号化では、領域分割と動き推定はそれぞれが互いに密接に関わりあっており、これらは符号量と SN 比が最適になるように行われるべきである。これは、画面中の各ブロックにどのように動きベクトルを配置すれば良いのかという最適化問題に置き換えることができる。そこで、本論文では最適な領域分割結果を与えるための GA のデータ型、各プロセスなどについて検討する。さらに、テスト画像を用いた計算機シミュレーション実験をし提案手法の有効性を示す。

Moving Picture Coding Based on Region Segmentation Using Genetic Algorithm

Koh'ichi TAKAGI Atsushi KOIKE Shuichi MATSUMOTO

KDD R&D Laboratories

2-1-15 Ohara Kamifukuoka-shi, Saitama 356-8502, Japan

abstract In this paper, we propose the moving picture coding based on region segmentation scheme using genetic algorithm(GA). In detail, moving picture coding based on region segmentation and motion estimation should be simultaneously done for decreasing coding bits and raising encoding S/N ratio. We study how to apply GA, for example, data format, initializing, each procedure, etc., for optimizing region segmentation.

1 はじめに

従来の矩形ブロック単位の動き補償予測符号化方式では、異なる動きが混在するブロックにおいて符号化効率が低下する問題が生じる。この問題を避けるために画像を性質の類似した領域ごとに分割して符号化する方式が注目されてきている。領域分割を動き補償予測符号化に適用することで、矩形ブロック分割による符号化方式の際に必要な領域形状を符号化する必要が生じる。しかし、一般的に予測誤差が減少するため、それにより領域形状符号量の増分を打ち消すことができると考えられている。

さて、動き補償予測誤差符号量を I_e 、動きベクトル符号量を I_v 、領域形状符号量を I_s とすると、制御ビット等を除いた全体の符号量 I は

$$I = I_e + I_v + I_s \quad (1)$$

で表される。

すなわち、動画の動き補償領域分割符号化方式は、画面を細かく分けた領域に対しある適当な動きベクトルを割り当てることで I と S/N を同時に最適にしていく最適化問題と言いかえることができる。なお画面を細かく分けて考えなければ領域分割の意味がなく、そのためには探索空間が馬鹿にならない。

遺伝的アルゴリズムはこのような大きな探索空間からある評価値を基準として最適な組合せを逐次的に見つけ出すのに適したアルゴリズムであり [2]、我々が本稿においてこれを適用した理由の一つとなっている。ただ、遺伝的アルゴリズムは染色体(ここでは遺伝子を持つ器と考えて下さい)上で解をどのように表現するかで最適化の性能が大きく変わる。そこがまず第一の問題となってくる。

そこで、本稿では第2章で我々が適用した遺伝的アルゴリズムのデータ型について述べ、さらに詳細なプロセスについても述べることにする。続いて第3章にそれを計算機シミュレーションした結果を示し、最後第4章でそれらをまとめる。

2 遺伝的アルゴリズムとその適用

2.1 遺伝的アルゴリズムとデータ型

遺伝的アルゴリズムは一般的に以下のような処理手順からなっている。

問題に応じて染色体を適当なデータ型に設定する。それに従い、初期染色体の組が設定される。そして染色体の評価、選択(増殖、淘汰)、交叉、突然変異を数回繰り返す、定められた終了条件により処理が終了する。

そこで、まず始めに染色体のデータ型の設定について検討する。染色体上への遺伝子の配置方法として以下の2方法を提案する。

2.1.1 境界線を基本としたもの

各小ブロックの境界線を遺伝子とする。具体的には画面を適当な大きさの細かなブロックに分け、各ブロック (B_i, B_j とする) の境界線を (i, j) で表す。またすべての (i, j) を任意の順番に並べ、その順番を $N(i, j)$ で表す。さらに、ベクトル L (これが染色体) を作成し、 (i, j) で統合が起こる起こらないを $N(i, j)$ ビット目の数値 0/1 で表すことにする。

画面全体で H (水平) \times V (垂直)、ブロックの大きさを $M \times M$ 画素とすると染色体の長さは図1のようなになる。

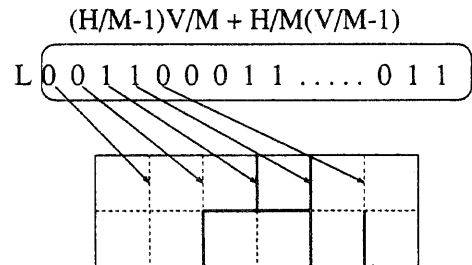


図1:境界線を基本としたもの

このやり方は染色体のビット数は莫大であるが、遺伝子自体は 0/1 しか取り得ない(すなわち染色体の種類は $2^{(\text{染色体の長さ})}$) ので 0/1 をどのように扱うかが鍵になる。さらに、各 (i, j) が相関を持ち密接に関わりあっていることも利用できる。

2.1.2 ベクトルマトリックスを基本にしたもの

各小ブロックに当てはめる動きベクトルを遺伝子とした。具体的には画面を適当な大きさの細かなブロックに分け、各ブロック (B_i) のベクトル v を適当に定める。なお、ブロックは2次元に分けられているので、染色体も2次元のものとなる。ここではベクトルマトリックスと呼ぶことにする。

この方法では染色体のビット数は前者より少ないが、遺伝子自体が動きベクトルの種類数 $Num(v)$ だけ存在するので染色体の種類数は

$$(H/M \times V/M)^{Num(v)}$$

となる。

以上2つの方法について述べたが、(実装の観点から) 本稿においては後者に対する検討結果についてのみ示すことにする。

次に全体的な処理手順について説明する。全体的な処理の中で「突然変異」は極値に落ち込んだ状態から脱出するためのものである。「突然変異」の後に「選択」を行ってしまうとその染色体が淘汰されてしまう可能性が高いと考えられる。そこで、まず始めに「評価」を行い、「選択」を行った後に「突然変異」、そして「交叉」を行うことにする。

これらの操作を以下簡単に述べる。

2.2 初期状態の設定 (initialization)

2.1.2のデータ構造を用い、下記のように染色体の初期状態を求めた。(なお、染色体は最初 N 個あるとする。)

1. $M \times M$ の小ブロックごとに、SSD(Sum of Square Difference) を計算済とする。隣接したブロックの組 (B_i, B_j) について

$$\begin{aligned} & \min_m SSD(B_i, v_m) + \min_n SSD(B_j, v_n) \\ & < \alpha \min_k \{SSD(B_i, v_k) + SSD(B_j, v_k)\} \end{aligned} \quad (\alpha < 1) \quad (2)$$

を満たす部分は統合されないものとする。(仮に統合されても、結果が良くなりそうにない部分を挙げる。)

2. (2) 式は平坦な領域に関して正確な値が出てこない可能性が高い。ほとんど動きがないにもかかわらず動きベクトルが異なると判断されてしまう場合、符号量の増大につながるばかりか画質も劣化する。そこで、動きベクトルの信頼度に着目する。具体的には文献 [3] を参照されたい。信頼度関数 $R(\theta)$ を

$$R(\theta) = \frac{a - \|p\| \cos(2\theta - \arg p)}{a^2 - \|p\|^2} \quad (3)$$

とおく。(なお、この式は任意方向 θ における推定誤差を近似したものである) ただし、 a, p に現画像の画素値 $f(x, y)$ の x, y に関する偏微分 f_x, f_y の離散的な場合の式をあてはめ、

$$a = \sum_{x,y} \frac{f_x^2 + f_y^2}{2} \quad (4)$$

$$p = \left(\sum_{x,y} \frac{f_x^2 - f_y^2}{2}, \sum_{x,y} f_x f_y \right) \quad (5)$$

とおく。さらに文献 [3] どおりに $R(\theta)$ の \max, \min に着目し

$$\max R(\theta) = \frac{1}{a - \|p\|} > T_1 \quad (6)$$

$$\frac{\max R(\theta)}{\min R(\theta)} = \frac{a + \|p\|}{a - \|p\|} > T_2 \quad (7)$$

(T_1, T_2 は閾値) となるブロックを抽出する。(なお参考までに、 a は物理的には輝度変化量を表し、 p はその方向性を表していると考えられる。) 先に述べたとおり、ここで抽出されたブロックは平坦な画像である可能性が高い為、別途動きベクトルを求めるべきである。また平坦であると言うことはまわりのブロックと動きを共にしている可能性が高いとも言えよう。ここでは、抽出されたブロックのまわりにあるブロックの動きベクトルの中で最適と考えられるものを使うことにする。

3. ブロックマッチングの結果をそのまま用いる。それによりブロックマッチングの手法を用いた場合を包含している。

2.3 染色体の評価 (evaluation)

符号量 (bit/pel) と S/N の関係を考慮して、最適な場合を適用できるように、評価式を設定する。具体的には符号量と S/N が線形関係にあると仮定し、

$$f(\text{染色体}) = k \cdot SNR - I/\text{pel} \quad (8)$$

(k : 定数, pel : 画素数) が大きくなる程よいものとする。なお、 $SNR = 20 \log_{10} \sqrt{\frac{\mu^2}{MSE}}$ (μ = 2(標本化ビット数) とする。

2.4 染色体の選択 (selection), 突然変異 (mutation)

染色体の選択, 突然変異に関しては以下の方法で行う。まず求められた評価値を参考にしてルーレット選択で N 本の染色体を選び出す。ここで, 評価値が似通うことが予想されるので, 評価値に対しスケーリングを行う必要がある。スケーリングには様々な方法が存在するが, 線形スケーリングを選択する [1]。まず, i 番目の染色体 C_i に対して求められた評価値 $f(C_i)$ ($i = 1, \dots, 2N$) を用いて $f_{\max}(= \max_i f(C_i))$ を求める。スケーリング関数

$$g(f(C_i)) = af(C_i) + b \quad (9)$$

に対し, ある定数 $c (> 1)$ を用いて

$$g(f_{\max}) = \overline{cg(f(C_i))} \quad (10)$$

$$\sum_i g(f(C_i)) = \sum_i f(C_i) \quad (11)$$

を満たす a, b を求める。これを式 (9) に当てはめ各染色体に対してスケーリングを行う。さらに, 評価値が最もよい染色体をエリート染色体と名付け, エリート保存方式 (最も評価値の高い染色体はそのまま次世代に残す) を採用することとする。

なお選び出す染色体は重複しても構わない。選ばれた染色体の中で複数出てきたものに関して (original の一つは残し, 残りに対して) 突然変異を施す。こうすることにより同じ染色体はできにくくなる。

また, 突然変異は一様に行うものとする。ただし, あまりにも現実からかけ離れた動きベクトルを適用するとそれにより符号量 (特に動き補償予測誤差符号量) が増加してしまい収束の速度を遅くさせることになりかねない。そこで, 少しでも収束の速度を早めるために各ブロックで SSD が小さい方から上位 m 位の動きベクトルを選びその中から任意の動きベクトルを選び出すことにする。

2.5 染色体の交叉 (crossover)

N 本の染色体を親として, 交叉を行う。交叉の方法としては色々考えられるが, 下記の方法について試した。

- 染色体中の任意の遺伝子を交叉させる。(normal な一様交叉)

- 染色体中に任意の矩形を作り交叉させる。(2次元版 2点交叉)
- 両親の染色体どうしを見比べて違う値が入っている部分を抽出し, 最も面積の広い部分を交叉させる。(図 2)

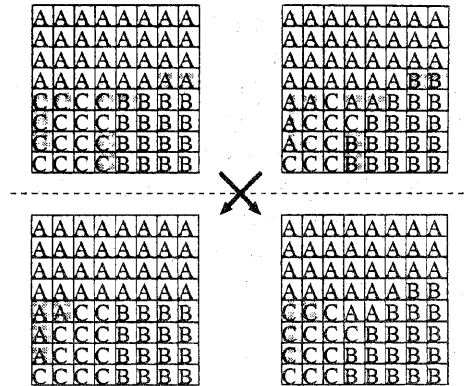


図 2: 両親の異なる部分が最大面積のブロックの集合を交叉

さらにこの後の染色体の評価のために親子とも次世代に引き継がれるものとする。(すなわち, 終了時 $2N$ 個になっている)

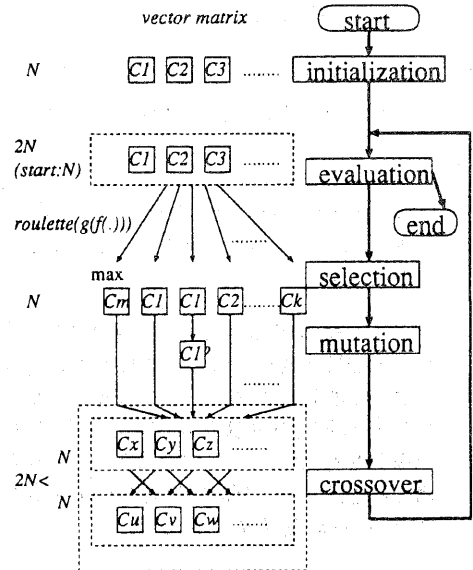


図 3: GA 流れ図 (GA flowchart)

2.6 終了条件

1. エリート染色体が定められた世代数変化しなかったとき。
2. 世代数が定められた数に達したとき。

のどちらか一つでも成り立つとき、ループ (世代交代) が終了するものとする。

全体的な処理の流れを図 3 に示す。

3 計算機シミュレーション

前章の要領で計算機シミュレーションを行った。使用したテスト動画は Cheerleaders (標準化ビット数: 8bit/pel, 画素数: 240 × 704) であり、パラメータに関しては表 1 のとおりに設定した。

最小ブロック画素数 ($M \times M$)	4 × 4
初期染色体数 (N)	10
α	0.7
突然変異率	3%
T_1	0.01
T_2	100
c (スケージング)	1.5
終了条件 1 (最適状態無変化世代数)	10
終了条件 2 (総世代数)	100

表 1: パラメータ設定

なお、符号量の計算に関しては、動き補償予測誤差量に DCT を施し、MPEG-2 における可変長符号を用いた。また動きベクトルは整数画素単位とした。初期状態 (2.2 節) は各方法により導出されたもの、および、それらを併用したものを使用した。

また、2.5 節における交叉の方法であるが、予備実験を行った結果、一様交叉、2 点交叉はあまり効果がないと考えられる為、最大面積交叉を行うものとする。

以上の条件で計算機シミュレーションを行った。結果を図 4 に示す。比較のため、全く同条件で行った (16 × 16 画素) ブロックマッチング (BM) 法の結果も同時に載せた。この結果から、様々なビットレートにおいて提案手法が BM 法を上回っていることがわかる。(このフィールド間予測において 0.3bpp でおよそ 0.2dB 程度、0.8bpp でおよそ 0.5dB 程度のゲイン。)

また、図 6 に示す動画を 16 × 16 画素で BM した結果を領域分割した結果を図 5 の左上に、提案法の領域分割の経過を同図の右上、左下、右下に示す。

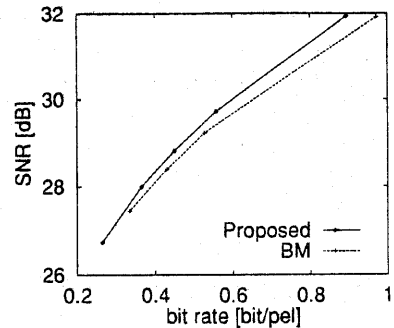


図 4: シミュレーション結果 (1 → 3field 予測)

4 おわりに

本稿では遺伝的アルゴリズム (GA) を使い領域分割・動き補償を同時並行的に処理する方式について提案した。

テスト画像を用いて計算機シミュレーションした結果提案手法は BM 法と比較して有効であることがわかった。

今後は、3 節で述べたパラメータを最適な方向に自動設定できるようにすること、GA の各処理の実施方法、動画の種類による提案法適用のノウハウなどについてさらなる検討をしていく予定である。

なお、本研究は通信・放送機構からの委託研究「ISDB 技術に関する研究開発」に基づき行われたものである。

謝辞

日頃より御教授頂く (株)KDD 研究所取締役所長村谷拓郎氏、同副所長鈴木健二氏に感謝致します。

参考文献

- [1] 安居院, 長尾: "ジェネティックアルゴリズム," 昭晃堂, 1993.
- [2] 荻原: "ニューロ・ファジィ・遺伝的アルゴリズム," 産業図書, 1994.
- [3] 吉田他: "動画の動きベクトルに対する信頼度関数とその応用," 信学論, Vol. J80.-D-II, No. 5, pp. 1192-1201, 1997.5.

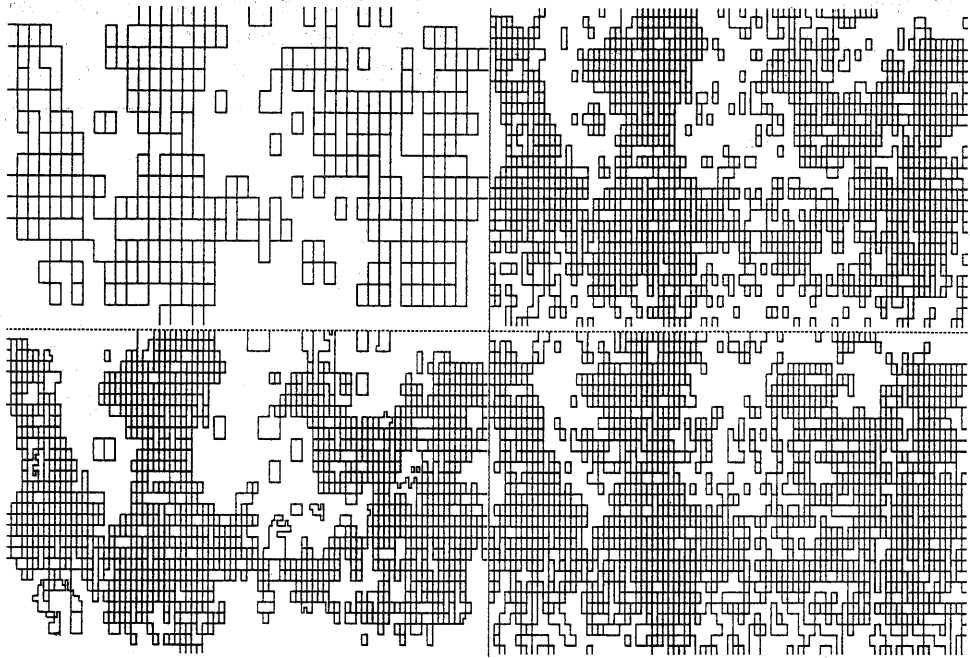


図 5: 領域分割結果 (左上: 16×16 BM, 右上, 左下, 右下: 提案法経過)

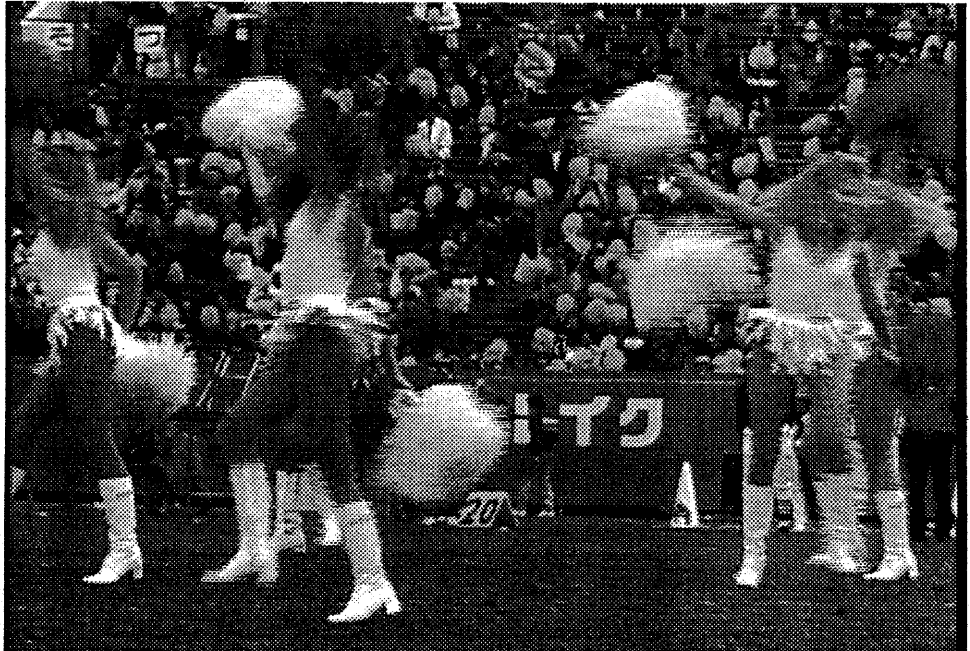


図 6: 原画像