

画像のアフィン変換動き補償予測に関する検討

A Study on Motion Compensated Prediction for Video Sequences based on Affine Transformation

水野 正吾 沢田 克敏

Shogo MIZUNO Katsutoshi SAWADA

ABSTRACT The motion compensated prediction is an essential technique for redundancy reduction coding of video sequences. This paper describes motion compensated prediction based on affine motion model where not only translation motion but also rotation and scaling motions are compensated. Computer simulation experiments using actual video sequences were carried out in order to clarify the relationship between affine parameters and coding performance. Prediction error entropy and affine parameter entropy were estimated for various parameter sets. The result have shown that half-pel translation is most effective for reduction of total coding entropy and that rotation and scaling are less effective compared with the translation.

1. はじめに

現在、動画像(テレビジョン画像)に対する高効率符号化において広く用いられている技術として動き補償予測(MCP: Motion Compensated Prediction)符号化方式^{1) 2)}がある。この方式では、現画像フレームの入力画像と前画像フレームの参照画像を比較して、動き検出(ME: Motion Estimation)を行い、検出した動き量を考慮して参照画像から入力画像を予測し、その動き量と予測誤差を伝送する。動き検出の中で主流になっているのが平行移動ブロックマッチング法^{1) 2)}である。この方式では入力画像を矩形ブロックに分割し、ブロック毎に、入力画像のブロック座標と同一位置の参照画像のブロック座標を中心にしてブロックを水平・垂直にシフトさせていき、入力画像と参照画像の間で最もマッチングの良い位置のシフト量を求め、これを動きベクトル(MCベクトル)として表現している。

しかし本来、動きには平行移動だけではなく、拡

大縮小(伸縮)・回転・変形といったものも考えられる。平行移動・伸縮・回転・変形といった幾何学的な変換のことをアフィン変換³⁾と呼ぶ。アフィン変換の動き補償への適用は、これまでもいくつかの報告^{4) 5) 6)}がある。しかしながらアフィン変換の各種のパラメータが動き補償予測符号化の特性にどのように寄与しているかについては十分に検討されていない。そこで本稿では、アフィン変換の各パラメータと符号化効率の関係について実画像データを用いた計算機実験により検討する。

2. アフィン変換動き補償予測符号化方式

2.1 画像のアフィン変換

アフィン変換を用いれば、平行移動・伸縮・回転・変形といった動きを幾何学的な変換で表現する事が可能^{7) 8)}である。変換式を式(1)に示す。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad \dots(1)$$

$$A_t = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

式(1)で (x, y) は変換前の画素の位置座標、 (x', y') は変換後の画素の位置座標、行列 A_t は回転・伸縮・変形変換を含む合成変換行列、 (t_x, t_y) は平行移動変換行列を表している。

実際にアフィン変換を行うときには、変形は考慮に入れない。なぜなら、伸縮・回転・平行移動は変換行列が決まっているが、変形には様々なパターンが考えられるため変換行列が決まっていないのでアフィン変換パラメータの算出がほとんど不可能になってしまうためである。以下では平行移動、および座標原点 $(0, 0)$ を中心とした回転・伸縮のみを考えることにする。このとき式(1)は式(2)に変形できる。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} C_x & 0 \\ 0 & C_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (2)$$

$$Rot = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

$$C_{xy} = \begin{pmatrix} C_x & 0 \\ 0 & C_y \end{pmatrix}$$

$$txy = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

式(2)の行列 Rot は回転パラメータ θ の回転変換行列、行列 C_{xy} は伸縮パラメータ C_x (水平方向)・ C_y (垂直方向)の伸縮変換行列、 txy は平行移動パラメータ t_x (水平方向)・ t_y (垂直方向)の平行移動変換行列を表している。 (x, y) と (x', y') は、式(1)と同様である。

アフィン変換をデジタル画像に対して行うときには、 (x, y) は整数値を取るが、式(2)から解るように (x', y') は多くの場合非整数値を取るため、その点における仮想的な画素値を整数位置の隣接画素から算出しなければならない。

図1に示すように (x', y') が非整数値の場合、隣接画素4点の画素値 $I(x_p, y_p), I(x_p + 1, y_p), I(x_p, y_p + 1), I(x_p + 1, y_p + 1)$ を参照し

(x', y') との距離に反比例した重み ($w_{x1}, w_{x2}, w_{y1}, w_{y2}$) 付けを行って補間値 $I(x', y')$ を算出する線形補間法を用いる。これを式(3)に示す。

$$\begin{aligned} I(x', y') &= w_{x2} w_{y2} I(x_p, y_p) \\ &+ w_{x1} w_{y2} I(x_p + 1, y_p) \\ &+ w_{x2} w_{y1} I(x_p, y_p + 1) \\ &+ w_{x1} w_{y1} I(x_p + 1, y_p + 1) \quad \dots (3) \end{aligned}$$

$$\begin{aligned} w_{x1} &= x' - x_p \\ w_{x2} &= 1.0 - w_{x1} \\ w_{y1} &= y' - y_p \\ w_{y2} &= 1.0 - w_{y1} \end{aligned}$$

$I(h, v)$: 座標 (h, v) の画素値
 x_p : x' を越えない最大整数
 y_p : y' を越えない最大整数

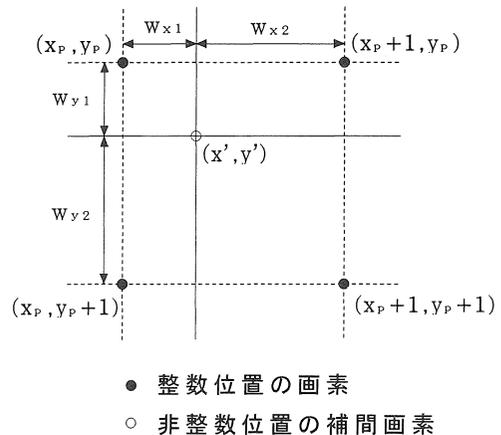


図1 非整数位置の画素値の線形補間法

2.2 アフィン変換動き補償予測方式

アフィン変換を動き補償予測方式に適用する場合、入力画像をブロック分割し、ブロック単位で式(2)を用いて変換を行う。まず、入力画像の各ブロック毎の中心を原点 $(0, 0)$ とし、ブロック内の各画素の座標 (x, y) を定める。また参照画像も入力画像の原点と同一座標を $(0, 0)$ とする。入力画像のブロック内の全ての画素の座標 (x, y) に対して同一のアフィン変換パラメータで、式(2)に従いアフィン変換を行い、座標 (x', y') を参照画像

の座標に対応させる。次に入力画像の座標 (x, y) における画素値 $I(x, y)$ と、参照画像の座標 (x', y') における画素値 $I(x', y')$ との間で予測誤差評価値 (例:画素値のブロック内差分絶対値和) を計算する。アフィン変換パラメータを探索範囲内で変化させ予測誤差評価値の最小となるパラメータを求め、これをそのブロックに対するアフィン変換動き補償予測のパラメータとする。

また、このときの座標 (x', y') における参照画像の画素値 $I(x', y')$ を座標 (x, y) における入力画像の画素値 $I(x, y)$ に対する予測値とする。アフィン変換動き補償予測方式で回転パラメータ $\theta = 0$ かつ伸縮パラメータ $C_x = C_y = 1$ を取る場合には平行移動ブロックマッチングと同等になる。回転パラメータ・伸縮パラメータが前記以外の場合、平行移動・回転・伸縮といった動きに対応したアフィン変換動き補償予測方式となる。

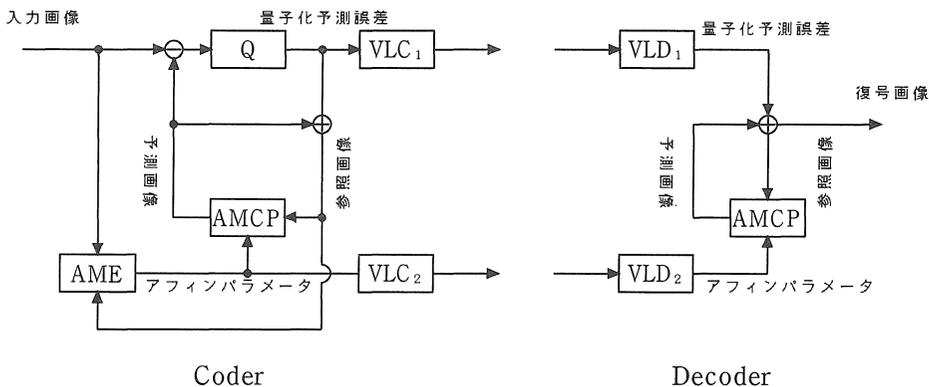
2.3 2段階アフィン変換パラメータ探索法

アフィン変換動き補償予測方式では、平行移動の2次元パラメータ (t_x, t_y) に加えて伸縮の2次元のパラメータ (C_x, C_y) と回転の1次元パラメータ (θ) の計5次元パラメータ $(t_x, t_y, C_x, C_y, \theta)$ の最適値を検出する必要がある。これらのパラメータの最適値を求めるためには、探索範囲

内でこれらのパラメータの全ての組み合わせについて調べる、いわゆる全探索をする必要がある。しかし、平行移動の2次元パラメータの全探索さえ演算量が膨大であることを考えると5次元パラメータの全探索は極めて非現実的である。

しかし、実際の画像中の物体の動きは、平行移動に占める割合が回転・伸縮に占める割合よりも非常に大きいので探索方法を平行移動パラメータの探索 (グローバルサーチ) と回転・伸縮パラメータの探索 (ローカルサーチ) の2段階に分けて探索⁴⁾すれば、演算量を大幅に削減することが可能であり、比較的正確にアフィン変換パラメータを検出することができる。さらにローカルサーチの時に平行移動パラメータをグローバルサーチよりも精度を細かくして探索 (微調整 D_x : 水平方向・ D_y : 垂直方向) することでより正確に平行移動のアフィン変換パラメータが検出できるようになる。

具体的に2段階アフィン変換パラメータ探索法では、まず物体の大まかな動きをとらえるために、グローバルサーチで平行移動パラメータ (t_x, t_y) の探索を行い、マッチングの良い位置を求める。このとき回転パラメータは $\theta = 0$ 、伸縮パラメータは $C_x = C_y = 1$ 、微調整パラメータは $D_x = D_y = 0$ に固定させておく。次にローカルサーチではグローバルサーチで求めた位置を中心とした微小空間で回転、伸縮、微調整パラメータの探索を行い、予測誤



AMCP: Affine Motion Compensated Prediction VLC : Variable Length Coding
 AME : Affine Motion Estimation VLD : Variable Length Decoding
 Q : Quantization

図2 アフィン変換動き補償予測符号化方式の構成

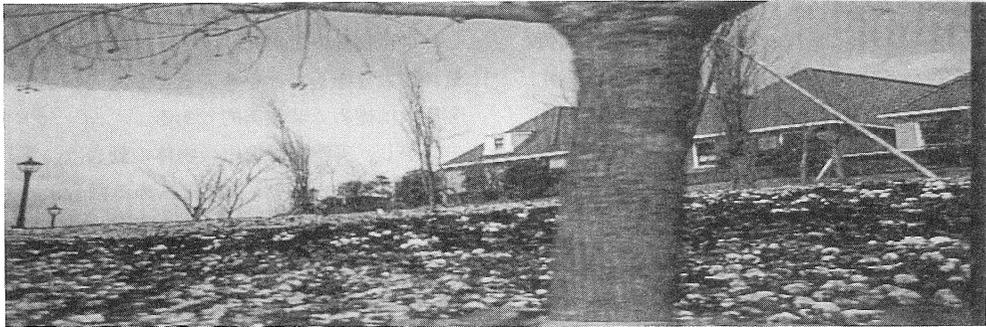


図3 実験に使用した画像 (Flower Garden 第2フレーム)

差評価値の最小値を取る各パラメータの組み合わせを求めて、それらを最適のアフィン変換パラメータとする。

2.4 アフィン変換動き補償予測符号化方式の構成

図2にアフィン変換動き補償予測符号化方式の構成を示す。この構成の送信側 (Coder) では AME で現フレーム (入力画像) と前フレームの復号画像 (参照画像) によりアフィン変換パラメータを検出し、AMCP でこのアフィン変換パラメータと参照画像を用いて動き補償予測 (予測画像の作成) を行い、入力画像と予測画像の差 (予測誤差) を Q で量子化し、量子化予測誤差、アフィン変換パラメータを別々に可変長符号化 (VLC_1 , VLC_2) する。受信側 (Decoder) では、前フレームの復号画像である参照画像とアフィン変換パラメータにより AMCP で動き補償予測 (予測画像の作成) を行い、予測画像に量子化予測誤差を加えることによって現フレームの復号画像を得る。ここで得られた現フレームの復号画像は次フレームに対する参照画像となる。

3. 計算機シミュレーション実験

3.1 実験条件

2.4 で述べた構成において、各アフィン変換パラメータが符号化効率に与える影響について実画像データによる計算機シミュレーション実験を行った。以下、計算機シミュレーション実験の際の各種条件

を述べる。

テスト画像には、"Flower Garden" 第1～6フレームの奇数フィールド (輝度信号: 720画素×240ライン) を用いた。図3にこの画像の第2フレームを示す。AMCブロックサイズは水平8画素×垂直8ライン、グローバルサーチの探索精度は表1、ローカルサーチの探索精度は表2・表3・表4に示す。グローバルサーチ + ローカルサーチを行った場合、表1と表2、表1と表3、表1と表4の3通りの組み合わせで符号化効率を比較した。

可変長符号化器 (VLC) では、理想的に符号化が行われるものとしてエントロピーを用いて符号化レートの評価を行った。

3.2 符号化特性の比較

図4・図5・図6に実験結果を示す。

ここで E_{MCP} はグローバルサーチにおける平行移動パラメータ情報とローカルサーチにおける回転・伸縮・微調整パラメータ情報のエントロピーの合計値、 E_{Err} は予測誤差エントロピー、 E_{All} は全符号化エントロピーをそれぞれ表している。プロットした点は使用画像6枚から得られた各エントロピーの平均値を表している。 $n1 = n2 = n3 = 0$ の点はグローバルサーチのみの場合で、それ以外は (グローバルサーチ) + (ローカルサーチ) の場合である。

図4に表1と表2の組み合わせの場合、すなわち、ローカルサーチで微調整パラメータ (D_x , D_y) のみ変化させた場合の符号化効率を示す。図4の縦軸の左側は E_{All} , E_{Err} のエントロピー、右側は

表1 平行移動パラメータ (t_x, t_y) の探索精度
($D_x = D_y = 0, C_x = C_y = 1, \theta = 0$)

範囲		精度	
t_x	t_y	t_x	t_y
± 16	± 8	1	1

表2 微調整パラメータ (D_x, D_y) の探索精度
($C_x = C_y = 1, \theta = 0$)

n1	範囲		精度	
	D_x	D_y	D_x	D_y
1	$\pm 1/2$	0	1/2	-
2	$\pm 3/4$	0	1/4	-
3	$\pm 7/8$	0	1/8	-
4	0	$\pm 1/2$	-	1/2
5	0	$\pm 3/4$	-	1/4
6	0	$\pm 7/8$	-	1/8
7	$\pm 1/2$	$\pm 1/2$	1/2	1/2
8	$\pm 3/4$	$\pm 3/4$	1/4	1/4
9	$\pm 7/8$	$\pm 7/8$	1/8	1/8

表3 伸縮パラメータ (C_x, C_y) の探索精度
($D_x = D_y = 0, \theta = 0$)

n2	範囲		精度	
	C_x	C_y	C_x	C_y
1	0.9~1.1	1	0.1	-
2	0.8~1.2	1	0.1	-
3	0.7~1.3	1	0.1	-
4	1	0.9~1.1	-	0.1
5	1	0.8~1.2	-	0.1
6	1	0.7~1.3	-	0.1
7	0.9~1.1	0.9~1.1	0.1	0.1
8	0.8~1.2	0.8~1.2	0.1	0.1
9	0.7~1.3	0.7~1.3	0.1	0.1

表4 回転パラメータ (θ) の探索精度 (単位: 度数)
($D_x = D_y = 0, C_x = C_y = 1$)

n3	範囲	精度	n3	範囲	精度
1	± 15	5	10	± 60	12
2	± 30	10	11	± 75	15
3	± 45	15	12	± 90	18
4	± 60	20	13	± 15	1
5	± 75	25	14	± 30	2
6	± 90	30	15	± 45	3
7	± 15	3	16	± 60	4
8	± 30	6	17	± 75	5
9	± 45	9	18	± 90	6

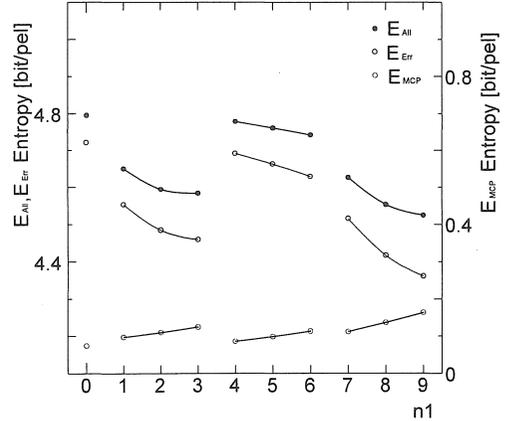


図4 微調整パラメータ (D_x, D_y) と符号化効率
($C_x = C_y = 1, \theta = 0$)

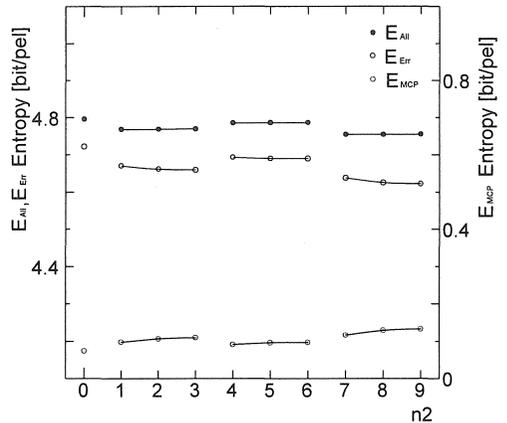


図5 伸縮パラメータ (C_x, C_y) と符号化効率
($D_x = D_y = 0, \theta = 0$)

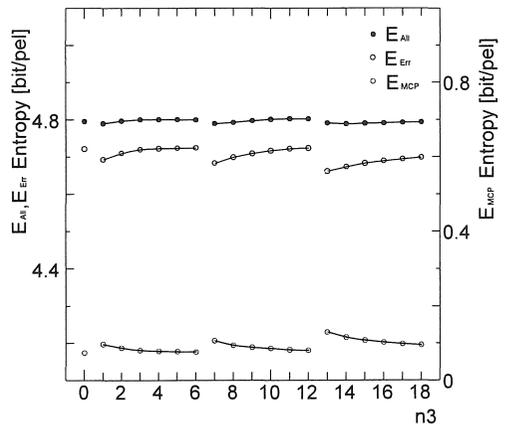


図6 回転パラメータ (θ) と符号化効率
($D_x = D_y = 0, C_x = C_y = 1$)

E_{MCP} のエントロピー、横軸は表2の探索精度 $n1$ をそれぞれ表している。

図5に表1と表3の組み合わせの場合、すなわち、ローカルサーチで伸縮パラメータ (C_x, C_y) のみ変化させた場合の符号化効率を示す。図5の縦軸は図4と同様であり、横軸は表3の探索精度 $n2$ を表している。

図6に、表1と表4の組み合わせの場合、すなわち、ローカルサーチで回転パラメータ (θ) のみ変化させた場合の符号化効率を示す。図6の縦軸は図4と同様であり、横軸は表4の探索精度 $n3$ を表している。

以上の図においては全符号化エントロピー E_{ALL} が低い値を取るほど符号化効率が良いことを示している。

回転・伸縮・微調整の3つをを比較すると、微調整が最も効果があり、次いで伸縮・回転の順に効果があった。

微調整については、 $n1 = 7$ の探索精度（探索範囲 $\pm 1/2 \cdot$ 探索精度 $1/2$ ）でも十分効果的である。探索精度を上げた場合 E_{Err} の減少量よりも E_{MCP} の増加量の方が低いので E_{All} は減少し効果があった。

伸縮については、探索精度を上げた場合 E_{Err} の減少量と E_{MCP} が増加量がほぼ同等なので E_{All} にはほとんど効果がなかった。

回転については探索範囲を狭くして、なおかつ精度を上げた方が僅かによい結果が出ている。しかし回転も伸縮と同様に、 E_{Err} の減少量と E_{MCP} が増加量がほぼ同等なので E_{All} にはほとんど効果がなかった。

結論として、微調整・伸縮・回転の順に効果があった。微調整は探索精度を上げた方が効果的であり、伸縮・回転は、探索精度を上げてほとんど効果はなかった。

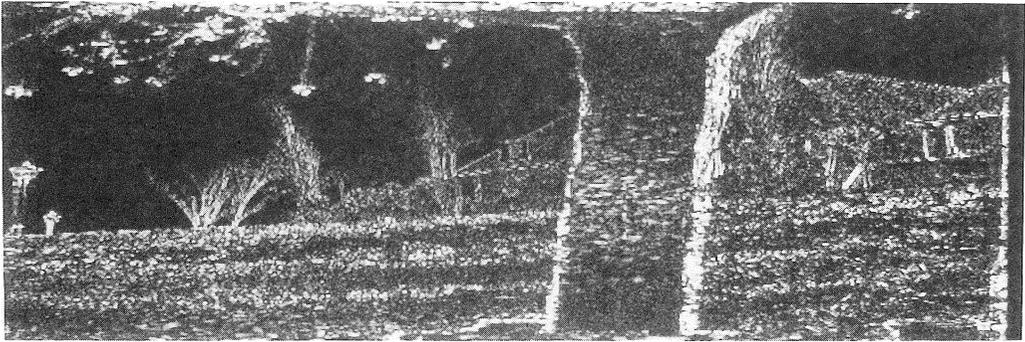


図7 Flower Garden 第3フレーム グローバルサーチの予測誤差画像
予測画像 SNR : 28.7561 [dB] 全符号化エントロピー : 4.6992 [bit/pel]

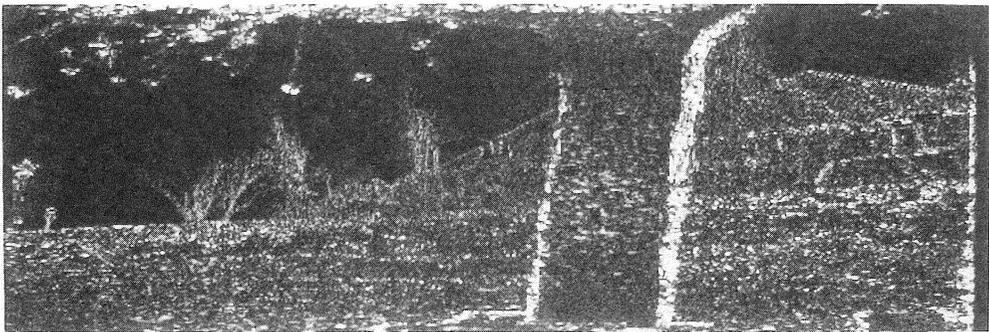


図8 Flower Garden 第3フレーム グローバル + ローカルサーチの予測誤差画像
予測画像 SNR : 30.7848 [dB] 全符号化エントロピー : 4.4619 [bit/pel]

3.3 予測誤差画像の比較

3.2の結果より、アフィン変換パラメータの組み合わせを最適値に設定したときのアフィン変換動き補償予測と従来の平行移動ブロックマッチング法の動き補償予測とで予測誤差画像の比較を行った。

従来の平行移動ブロックマッチング法と同等であるグローバルサーチのみの場合の予測誤差画像を図7に示す。アフィン変換動き補償予測であるグローバル+ローカルサーチ ($C_x \cdot C_y$: 範囲 0.8~1.2 精度 0.1、 θ : 範囲 $\pm 15^\circ$ 精度 3° 、 $D_x \cdot D_y$: 範囲 $\pm 3/4$ 精度 $1/4$) を行った場合の予測誤差画像を図8に示す。これらの図は予測誤差の絶対値を8倍して示している。図7よりも図8の方が予測誤差が低減(黒い部分の面積が大きい)しているので、アフィン変換動き補償予測方式は効果的であることがわかる。

4. むすび

本稿では、アフィン変換を動き補償予測に適用する方法、およびその際の効率的な探索法について述べ、各アフィン変換パラメータに対する符号化効率について計算機シミュレーション実験により比較検討した。アフィン変換パラメータは、微調整が最も効果があり、続いて伸縮・回転の順に効果があった。微調整は探索精度を上げた方が効果的であり、伸縮・回転は、探索精度を上げてほとんど効果はないと

いう結果が得られた。

今後の課題としては、矩形ブロックに限定しない任意形状領域でのアフィン変換動き補償予測、アフィン変換動き補償予測を行う際の演算量の一層の削減などが挙げらる。

文献

- 1) 原島博 監修: '画像情報圧縮', オーム社, 1994
- 2) 堀内司朗, 有村一郎 監修: '画像圧縮技術のはなし', 工業調査会, 1993
- 3) 伊原信一郎 河田敬義 著: '線形空間アフィン幾何', 岩波基礎数学選書, 1990
- 4) 如澤裕尚: 'アフィン変換を用いた動き補償予測に関する検討', 電子情報通信学会技術研究報告, IE94-37, 1994 7
- 5) 山口尚吾, 駄竹健志: 'アフィン変換を用いた一般化動き補償に関する一検討', 画像符号化シンポジウム (PCSJ93 No.8), 1993 10
- 6) 如澤裕尚, 小寺博: 'アフィン変換を用いた任意形状領域のMCに関する検討', 1994 信学春全大, D-332
- 7) 高木幹雄, 下田陽久 監修: '画像解析ハンドブック', 東京大学出版会, 1991
- 8) 田村秀幸 編: 'コンピュータ画像処理・応用実践編 1', 総研出版, 1990

(受理 平成8年3月19日)