卒業論文

# オブジェクトに着目した映像符号化に関する研究

東北大学 工学部 情報知能システム総合学科
大町研究室 ４年
石森　亮輔

# 目 次

# 図 目 次

# 第1章　序論

## 1.1　背景

現在，インターネットの普及により，誰でも簡単に情報を入手することができる．その中でも映像や画像は直感的にその場の様子や状況を伝えることができるメディアであり，情報を受け取る側の理解の助けにつながっている．

しかし，映像，画像は文字情報に比べて容量が大きいという問題点がある．未だに爪痕が残る2011年3月11日に日本を襲った東日本大震災など，災害が発生した際にはその状況伝達や，安否確認のために非常に多くの通信が行われることになる．そのため，通信トラフィックが増大し，処理能力を超える通信が集中すると，満足に通信を行うことができなくなってしまう．

そのような状況下で映像を送信する場合，ある程度容量圧縮を行ってから送信することが重要となる．だが，圧縮を行うことは同時に情報の損失の可能性もあり，重要な情報が損失されてしまえば，その画像には価値がなくなってしまう．

## 1.2　目的

以上の背景から本研究の目的は，重要な情報を損失させない圧縮を行うことである．本研究では，各オブジェクトを重要な情報と位置づけ，輪郭などがぼやけてしまわないように圧縮を行う手法の開発を目指す．

そこで，本研究では画像抽象化法に着目した．画像抽象化法とは重要な情報を減らしつつ画像の複雑さを減らす画像加工技術である．この画像抽象化に着目した理由として，画像を単純化することにより連続したピクセルで同じ画素値が続くことが多くなり，ハフマンコードで符号化した際により圧縮が図れる．ハフマンコードとは現在主要に用いられている画像ファイル形式であるJPEG,PNG等で使われる符号化法であり，より多く出てくるデータに対し短い符号を割り当てることにより圧縮を行っているものである．

さらに抽象化のもう一つの利点として，処理回数を変更することによって抽象度を変更することが可能なので，その目的に応じてオブジェクト毎の圧縮レベルを調整することができることである．

以上の理由により，画像抽象化を用いてオブジェクトの特徴を保持しつつ，高効率に符号化する手法の研究を行った．

## 1.3　本論文の構成

**第 1 章**　本研究の背景・目的についてのべた.
**第 2 章**　2 つの提案手法について述べる.
**第 3 章**　実験及びその考察について述べる.
**第 4 章**　まとめと今後の課題について述べる.

# 第2章　手法

## 2.1　概要

本研究では画像抽象化法について 2 種類の手法を用いた．一つはバイラテラルフィルタを用いる手法，もう一つは ETF，CMCF，ショックフィルタという 3 つの処理を用いて等輝度線の曲率を滑らかにしていく手法である．この両手法を比較することによって，符号化するのに有利な手法を検討した．なお，便宜的に前者をＡの手法，後者をＢの手法と呼称する．

## 2.2　Ａの手法

バイラテラルフィルター―はガウシアンフィルタでノイズを除去する際，輪郭もぼやけてしまうのを解消しようとしたアルゴリズムである．ガウシアンのカーネルに対し，中心の輝度値に近いものほど強い重みをつけることでエッジの損失を防ぐことができる．ガウシアンフィルタの式を 2.1 式，バイラテラルフィルタの式を 2.2 式で示した．

$$G(i,j) = \frac{\sum n = -w^{w} \sum_{m=-w}^{w} f(i+m, j+m) exp(\frac{m^2+n^2}{2\sigma_1^2})}{\sum_{n=-w}^{w} \sum_{m=-w}^{w} exp(-m^2+n^2/2\sigma_1^2)} \tag{2.1}$$

$$g(i,j) = \frac{\sum_{n=-w}^{w} \sum_{m=-w}^{w} f(i+m, j+m) exp(-\frac{m^2+n^2}{2\sigma_1^2}) exp(-\frac{(f(i,j)-f(i+m,j+n))^2}{2\sigma_2^2})}{\sum_{n=-w}^{w} \sum_{m=-w}^{w} exp(-\frac{m^2+n^2}{2\sigma_1^2}) exp(-\frac{(f(i,j)-f(i+m,j+n))^2}{2\sigma_2^2})} \tag{2.2}$$

f(i,j) は処理前の画像データ g(i,j) が処理後の画像データである．$w$ がカーネルサイズ，$\sigma_1$ はガウシアンフィルタの係数，$\sigma_2$ は輝度差の制御を表しており，$\sigma_2$ を大きくしていくとガウシアンフィルタの処理に近づき，逆に小さくしすぎるとノイズ除去効果が弱くなる．

以下にガウシアンフィルタ，およびバイラテラルフィルタの処理の例を示す．ここで $w = 7$ , $\sigma_1 = 50$ , $\sigma_2 = 25$ とした．

## 2.3　Ｂの手法

### 2.3.1　概要

この手法では，ETF,CMCF, ショックフィルタの 3 つの処理を反復的に行うことにより抽象化を実現する．この手法のフローチャートを図 2.2 に示した．以下にそれぞれの処理について記す．



(a) 原画像　(b) ガウシアンフィルタ処理例　(c) バイラテラルフィルタ処理例

図 2.1: A の手法の処理例

### 2.3.2 ETF(Edge Tangent Flow)

ETF(Edge Tangent Flow)[1] は，画像中のエッジ方向を向いたベクトル場を生成し，その流れを滑らかにするという手法である．ETF の手順は次のとおりである．

**i.** 画像中の各ピクセル $x = (x, y)$ に対して，輝度値の勾配ベクトル $g(x) = \nabla I(x)$ を計算する．

**ii.** $g(x)$ を 90 度回転し，$t(x)$ とする．

**iii.** 以下の式により反復的に $t(x)$ を更新する．

$$t^n ew(x) = 1/k \sum_{y \in \Omega(x)} t^{cur}(y)\omega_m(x, y)\omega_d(x, y) \tag{2.3}$$

$g(x)$ は輝度値が最も急な部分を向いているので，これを 90 度回転させる $t(x)$ はエッジに沿った方向になる．

ベクトル場を可視化したものが次の画像のようになる。

### 2.3.3 MCF(Mean Curvature Flow)

MCF(Mean Curvature Flow)[2] は，B の手法で用いられている CMCF(Constrained MCF) の元となる手法で，画像中の等輝度線の曲率を滑らかにする処理を行う．

等輝度線の曲率が小さい部分ではあまり輝度値を変化させないが，等輝度線が複雑な部分でその曲率が小さくなるように変化させる．

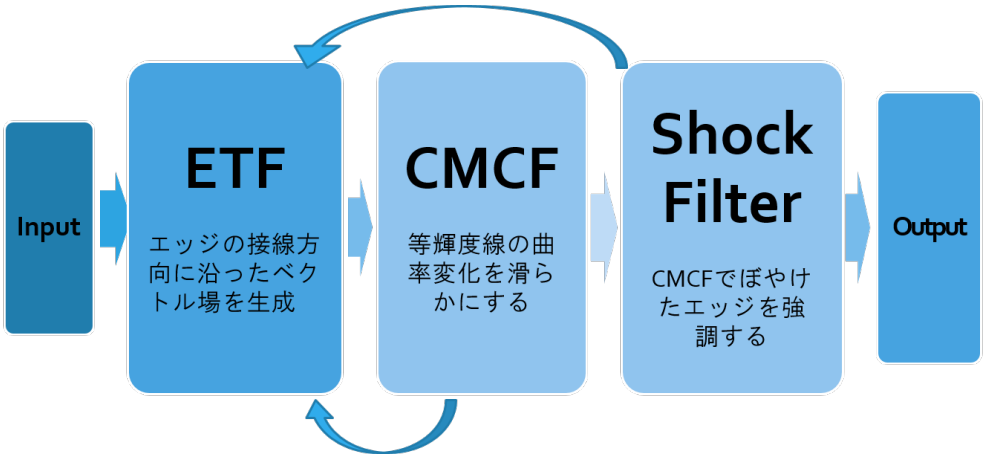
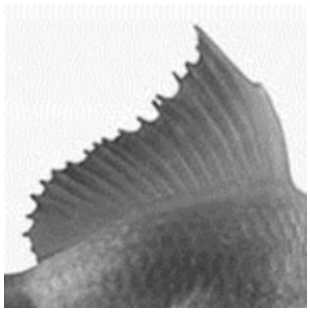
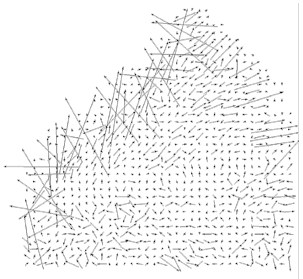



図 2.2: B の手法のフローチャート



(a) 原画像



(b) 輝度値の勾配ベクトル

(c)90 度回転、正規化後

(d) スムージング結果

図 2.3: ETF の処理イメージ

MCF は画像に対し反復的に処理を行い，一回の適用による各々のピクセルの輝度値変化量 $I_t$ は次のようになる．

$$I_t = \frac{dI}{dt} = \kappa|\nabla I| \tag{2.4}$$

ここで，$I(x)$ はピクセルの輝度値, $x = (x, y)$ はピクセルの画像中の位置,$\kappa$等輝度線の曲率を意味し,$\kappa$は次の式で求められることが分かっている．

$$\kappa = \frac{I_x^2 I_{yy} - 2I_x I_y I_{xy} + I_y^2 I_{xx}}{(I_x^2 + I_y^2)^{3/2}} \tag{2.5}$$

$\kappa$ の符号は，等輝度線の谷線周辺で正，尾根線周辺で負となる．輝度値は，$\kappa > 0$ で増加，$\kappa < 0$ で減少する．曲率の大きさに比例した量だけ，谷線周辺で輝度値を変化させ，尾根線周辺で輝度値を減少させることで，等輝度線の曲率が滑らかになる．

図 2.4 は，MCF の結果と等輝度線の形の一例である．左上が初期入力画像で，下に行くほど反復回数が大きい．左が画像，右画等輝度線の一部を可視化したものである．

### 2.3.4 CMCF(Constrained MCF)

CMCF は，ETF をで作ったベクトル場の重みを MCF に付加したものである．局所的に MCF の単純変化速度を変化させることによって，特徴を保持し，強調するようにしたものである．

MCF(式 2.4) に制約条件として，速度調整変数 s(x) を取り入れる．

$$I_t = dI/dt = s \cdot \kappa|\nabla I| s(x) = (1 - r) + r \cdot |t(x) \cdot \nabla I(x)^{\perp}| \tag{2.6}$$

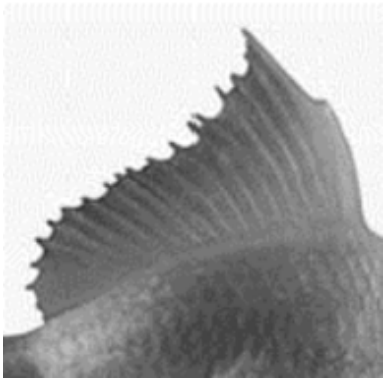
ここで, $r$(0〜1) は制約の強さで, $r = 0$ で $s(x) = 1$ となり, $MCF$ と同じ式になる．

$\nabla I(x)^{\perp}$ は勾配ベクトル $\nabla I(x)$ に垂直なベクトルで，すなわちエッジに沿った方向 (接線方向) を向いたベクトルとなる．t(x) は，ETF によって滑らかになったベクトルで，場所 x での望ましい方向といえる．これらの，元画像でのエッジ接線方向ベクトルと，それを滑らかにしたベクトルの内積によって s(x) は計算される．また，これらは単位ベクトルとする．二つのベクトルが平行ならば，s(x) は 1(制約なし)，すなわち単純化速度は速く，垂直に近づくにつれて s(x) は小さくなり，単純化速度は遅くなる．

### 2.3.5 ショックフィルター

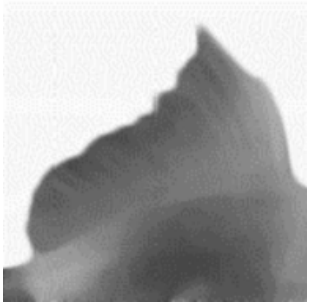
CMCF によってある程度は特徴を保持できるものの，エッジをぼやけさせてしまう．ぼやけたエッジを強調するために，ショックフィルターを用いる．各々のピクセルの輝度値変化量 $I_t$ は次のようになる．

微分演算はノイズを強調してしまうため，ガウシアンフィルタ $G_\sigma$ による平滑化を用いている．$(G_\sigma * I) = 0$ のゼロクロスの部分が輝度値変化が最も急な部分であるので，そこをエッジであるとする．$(G_\sigma * I) < 0$ で $I_t$ は正，$(G_\sigma * I) = 0$ で $I_t$ は負となる．この処理により $(G_\sigma * I) = 0$ 付近の輝度値変化が大きくなり，エッジが強調される．



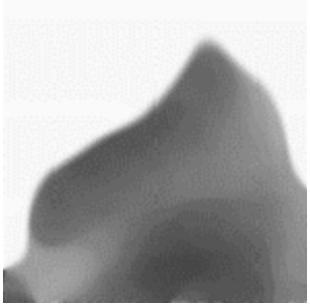
(a) 原画像

(b)a. の等輝度線の一部を可視化したもの



(c)MCF10 回適用後

(d) c . の等輝度線の一部を可視化したもの



(e)MCF60 回適用後

(f)e. の等輝度線の一部を可視化したもの

図 2.4: ETF の処理イメージ

# 第3章　実験

## 3.1　予備実験

B の手法のそれぞれの処理の効果を確認するため、予備実験を行った。

### 3.1.1　実験環境

予備実験において、条件を変えるとき以外は以下の環境で実験を行った。

(1) PC

- Windows 7 Home Premium
- Corei5-2410M
- メモリ：4GB

(2) 変数を固定するときのループ回数

- ETF：3 回
- CMCF：10 回
- ショックフィルター：CMCF5 回毎

(3) 使用画像

- cat.jpg

- 225 * 225 pixel

### 3.1.2　ETF の効果の検討

ETF のループ回数を 0,1,3,5,10 回と変化させたときの画像の見え方を比較し、その効果を検討した。

結果は以下のとおりである。ETF のループ回数が増えるにつれて、より特徴を残した画像に変換できていることが確認できた。

(a)ETF:1 回　(b)ETF:3 回

(c)ETF:5 回　(d)ETF:10 回

図 3.1: ETF の効果の検討

### 3.1.3　CMCF の効果の検討

CMCF のループ回数を 1,5,10,30 回と変えたときの画像抽象化の度合いを確認した。

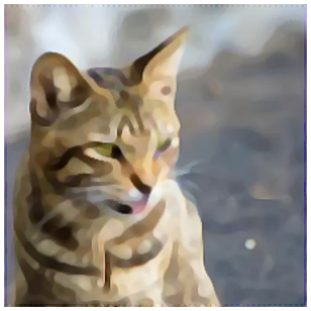
結果を 3.1.3 に示した。画像から、ループ回数を増やすほどより滑らかな画像を作ることができた。ただし、ループ回数が多くなりすぎると特徴が失われてしまった。

### 3.1.4　ショックフィルターの効果の検討

ショックフィルターをかけた場合とかけない場合で画像の見え方を比較した。

(a)CMCF:1 回

(b)CMCF:5 回



(c)CMCF:10 回

(d)CMCF:30 回

図 3.2: CMCF の効果の検討

(a) ショックフィルター有り

(b) ショックフィルター無し

図 3.3: ショックフィルターの効果の検討

結果が 3.3 である。ショックフィルターをかけた画像の方が、エッジがくっきりしているのが分かる。

以上予備実験から、Bの手法それぞれの処理において効果が確認できた。

## 3.2 抽象化の圧縮効果の検討

画像抽象化適用前後の画像のデータ量を調べることで、画像抽象化が映像圧縮に効果があるかを検討した。この実験では、抽象化自体の圧縮効果の有無の検討を行うために非圧縮 tiff 画像に対し画像抽象化の処理を行った後、ハフマンコードでの圧縮が可能な可逆圧縮の PNG 画像に変換して、データ量を比較した。なお、実験で用いた画像は 3.4 に示す 9 枚であり、その圧縮率の平均を求めた。

結果が図 3.5 である。この図から、抽象度を上げることによって画像を圧縮する効果があることが確認できた。

## 3.3 画質評価

### 3.3.1 客観的評価指標による評価

本論文では客観的評価をするにあたって、SSIM という評価指標を用いた。SSIM の式を 3.1 に示す。

$$SSIM(x,y) = (2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)/(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2) \tag{3.1}$$

ここで，$x$, $y$ は比較する輝度画像，$\mu_x$,$\mu_y$ は輝度値の平均、$\sigma_x$,$\sigma_y$ は画像の輝度の分散，$\sigma_xy$ は共分散,$c_1$,$c_2$ は，輝度値の平均や輝度値の分散が小さすぎるときに異常な値を示さないようにするための定数であり, 実験的に $c1 = 6.5025$,$c2 = 58.5225$ とした.

3.5 で用いた画像について、抽象化前後の画像に対し SSIM で評価し、その平均を求めた。

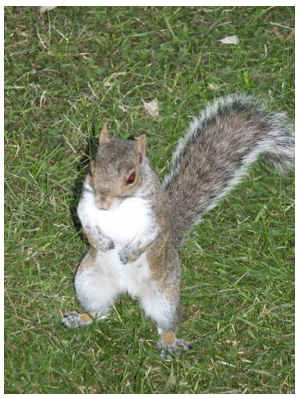
結果が図 3.6 である。ループ回数が増えるにつれ、SSIM の値も減少してしまうことが確認できた。

この結果を受けると、抽象化の処理を繰り返すたびに画像が劣化しているため、不適であるという結論になってしまう。しかし、本手法は画像を加工する技術であるため、元画像と比べて変化があるのは当然であるから、評価法があまりふさわしくないという結論に至った。

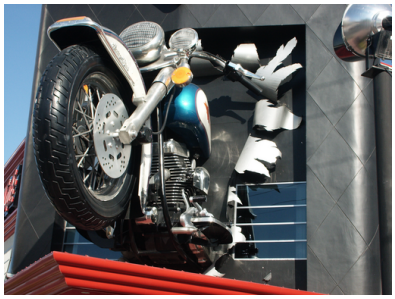
### 3.3.2 主観評価による画像の見え方の比較

#### 実験手順

次に、送信可能なデータ量が制限されたことを想定して、抽象化前後の画像を同データ量になるように圧縮を行い、その画像の見え方を比較した。まず、非圧縮 TIFF 画像の元画像に対して画像抽象化の処理を行う。次に、抽象化前後の画像を JPEG に変換して同じデータ量になるよう



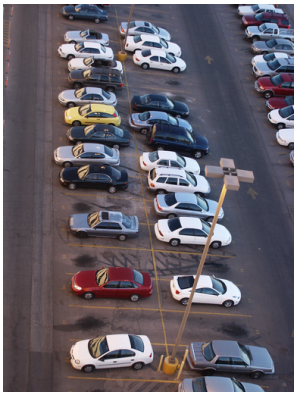


図 3.4: 使用した画像

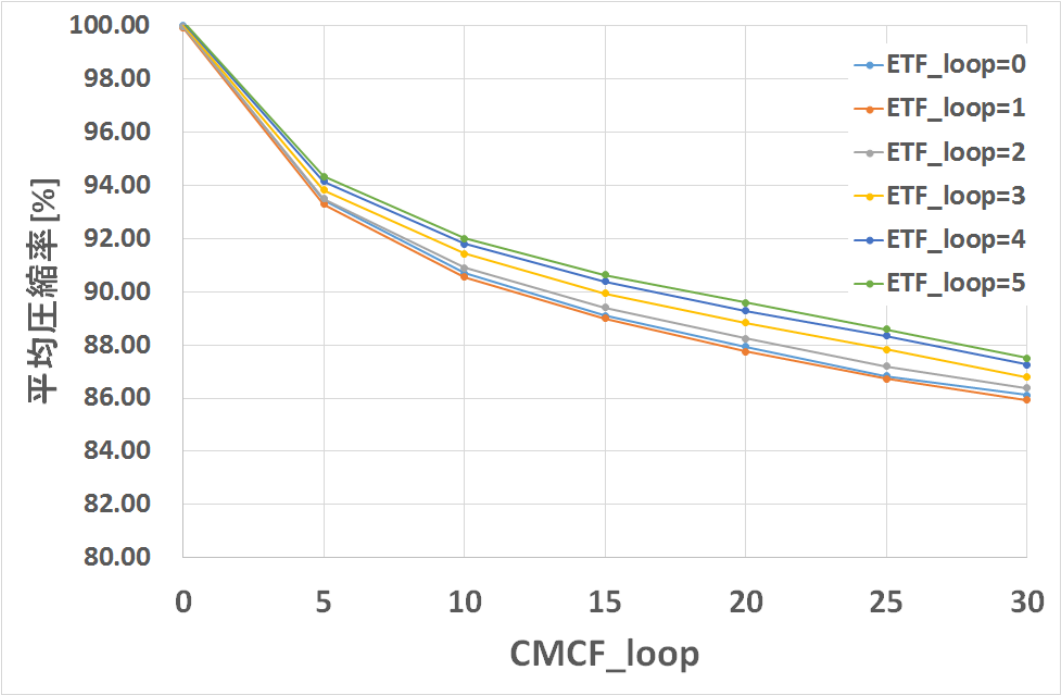



図 3.5: 平均圧縮率

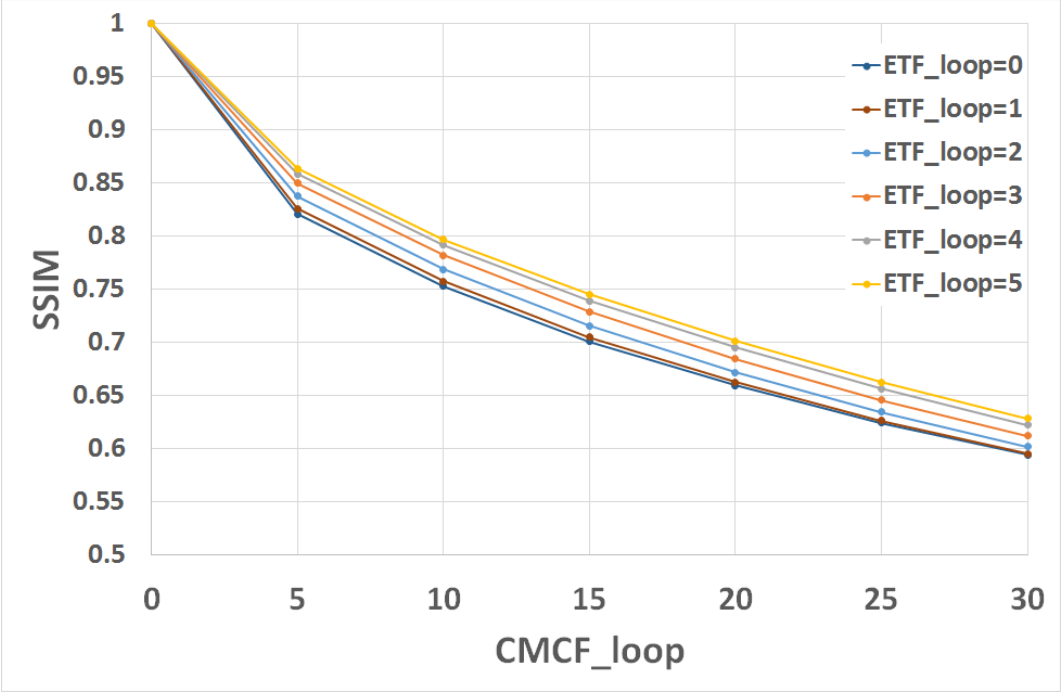



図 3.6: SSIM による客観的評価

に圧縮を行う。このようにして、元画像をそのまま圧縮したものと、A、Bの手法でそれぞれ抽象化した後に圧縮した3種類の画像を比較した。

#### 実験条件

実験条件は次のとおりである。なお、ここで設定した各パラメータは実験的に結果が良かったものを使用している。

(1) 共通条件

- 元画像データ量 (TIFF): 90.8KB
- 元画像データ量 (JPEG): 31.7KB
- 圧縮後にそろえるデータ量：3KB

(2) Aの手法のパラメータ

- $\omega$：8
- $\sigma_1$：50
- $\sigma_2$：25
- 繰り返し回数：20回

(3) Bの手法のパラメータ

- ETF：5回
- CMCF：15回
- ショックフィルター：CMCF5回毎

#### 実験結果

元画像と、実験により圧縮された3種類の画像を図2.4に示した。この結果を見ると、そのまま圧縮をした時よりも、抽象化の処理を行った後に抽象化した二つの画像の方がノイズが減っているのが分かる。このことから、画像抽象化法はより特徴を残した画像圧縮に効果があることが確認できた。

## 3.4 両手法の比較

Aの手法とBの手法で画像抽象化を行う。この際、画像の劣化がない可逆圧縮であるPNG画像に変換し、その画像の見え方、データ量、処理時間の観点から両者を比較した。


そのまま圧縮


A の手法適用後に圧縮


B の手法適用後に圧縮

図 3.7: 画質の主観評価

### 3.4.1 実験条件

実験の際のパラメータは以下のものを用いた。なお、ここで用いたパラメータも実験的に結果が良かったものを使用している。

(1) A の手法のパラメータ

- $\omega$ : 8
- $\sigma_1$ : 50
- $\sigma_2$ : 25

以上の条件で 20 回ループさせた。

(2) B の手法のパラメータ

- ETF ループ数 : 5 回
- CMCF ループ数 : 15 回
- Shock Filter の処理頻度 : CMCF5 回毎

(3) 用いた画像

図 3.8: 手法の比較

| | A の手法 | B の手法 |
|---|---|---|
| 画像 |  |  |
| データ量 | 82.4KB | 104KB |
| 処理時間 | 小 | 大 |

### 3.4.2 実験結果

実験結果を表 3.8 に示した。まず画像に関しては、A の手法の画像はノイズをよく削減で来ているが、背景とオブジェクトがあまり分けられておらず特に背景と色が近いオブジェクトは同化してしまっている。一方 B の手法の画像では色が近いものでも、はっきりオブジェクト毎に分かれている。しかし各オブジェクトを比べると、B の手法の方がより抽象化されて画質が劣化してしまっている。

また、データ量は A の手法の方が小さく、処理時間も A の手法の方が短かった。

以上の比較から、本稿での目的である抽象化を用いた高効率符号化においては A の手法がふさわしく、領域分割などで背景とオブジェクトを分けるような手法に応用する際は B の手法を用いるのが適しているという結論になった。

# 第4章　まとめと今後の課題

## 4.1　まとめ

本稿では、オブジェクトに着目した映像符号化のために、画像抽象化を用いた画像圧縮を行った。抽象化によりオブジェクトが単純化され、同時にノイズを削減することができた。輝度値変化が滑らかになったことで、ハフマン符号化による効果がより得られ、抽象化自体でデータ量を圧縮することができた。

また画質評価の点では、PSNR や SSIM の数値はかなり減少してしまうが、見た目上はその数値以上に画質を維持することができた。その際原画像をそのまま圧縮したものに比べ、提案手法ではノイズを削減することができた。

A の手法と B の手法を比較すると、高効率符号化には A の手法がより適しているという結果になった。しかし、B の手法の方がオブジェクトと背景をよく分離できていることから、領域分割などに応用する際は B の手法を用いた方が良い結果が期待できると思われる。

## 4.2　今後の課題

A,B の手法ともに実験的にパラメータを決めたが、これを自動化することで、画像毎により最適化された手法を適用できるようにしたい。

また、A,B の手法のいいところをうまく取り入れたような手法ができれば応用にも使用できる、処理時間の短い手法が実現できると考えている。

謝辞

本稿を執筆するに当たり、多大なご指導を頂くとともに、本研究の研究機会を与えてくださった、東北大学大学院工学研究科大町真一郎教授に深く感謝いたします。

また、多くの助言や指摘を頂いた東北大学大学院工学研究科菅谷至寛助教、並びに東北大学大学院工学研究科大町研究室研究員宮崎智さんに心より感謝致します。

最後に、日頃研究生活において多くのアドバイスを頂いた、大町研究室の皆様に深く感謝いたします。

# 関連図書

[1] C Tomasi, R Manduchi :

"Bilateral filtering for gray and color images",

Computer Vision, 1998. Sixth International Conference on, pp.839-846 (Jan 1998).

[2] H. Kang and S. Lee :

"Shape-simplifying Image Abstraction",

Computer Graphics Forum, vol. 27, no. 7, pp.1773-1780 (2008).

[3] H. Kang and S.Lee and C.K.Chui :

"Coherent line drawing "

In Proc. Non-photorealistic Animetion and RenderingCoherent line drawing, pp.43-50 (2007).