

グラフスペクトル法を利用した画像圧縮

Image compression with spectral methods

山田 真義†
Masayoshi Yamada

末松 伸朗†
Nobuo Suematsu

林 朗†
Akira Hayashi

1 はじめに

画像圧縮で用いられる変換符号化は、直交変換を用いて無相関化された画像データを量子化、符号化することで圧縮を行っている。直交変換には画像や動画画像圧縮に広く用いられている離散コサイン変換 (DCT)、そして KLT (Karhunen-Loeve Transform) 等が挙げられる。KLT は情報源の共分散行列をもとに変換行列を生成するため、相関の強いガウシアン情報源に対する最良の変換である。しかし、性能を上げるためには GMM (Gaussian Mixture Model) をデータ分布に当てはめなければならない。一方 DCT は KLT より性能は若干劣るが、変換行列が情報源と独立であることから、固有値分解等を行う KLT と比べ計算時間そして記憶サイズ的大幅な削減が可能である。

本稿では、パターン認識の分野で広く研究されているグラフスペクトル法と、既に提案されている KLT に基づく画像圧縮 [1] とを組み合わせた手法を構築し、画像圧縮性能の向上を目指す。なお、本稿では固定レート変換符号化器のみを取り扱う。

2 グラフスペクトル法を用いた画像圧縮

画像等の実世界に存在するデータは高次元になるほどその構造はより複雑となる。よって高次元空間のデータに対して GMM を当てはめたとしてもデータ構造が複雑であることから良い当てはめを行うことは困難であると考えられる。そこで本稿では、高次元空間のデータを一度低次元空間に埋め込み、その埋め込まれたデータ点に対して GMM を当てはめる。こうすることで、複雑な高次元空間のデータに対する GMM の当てはめよりも良い結果を得ることが期待される。

本稿における画像圧縮手法は以下の3つの部分からなる。

- A. 訓練画像を基に画像圧縮コーデックを作成する
1. 訓練画像にグラフスペクトル法を適用する。(2.1)
 2. 1.の結果に GMM を当てはめ、元の空間での共分散、平均、混合率を計算する(2.1)
 3. 2.の結果を基に各コンポーネントで KLT を行う。(2.2)
 4. 3.の結果を基に各コンポーネント、そしてブロックの各要素へのビット割当を行う。(2.3)
 5. 4.の結果を基に各コンポーネントの各ブロックにおけるスカラー量子化器を作成する。

- B. A. で作成したコーデックを基に、圧縮対象となる新たな画像を圧縮する。(2.4)
- C. A. で作成したコーデックを基に、圧縮データを復元する。

以下では、A. に重点を置き本稿の手法について説明する。

2.1 グラフスペクトル法

本稿では多くのグラフスペクトル法の中で、高次元空間内のデータ点の近傍からなる局所的なデータ構造を保存し低次元空間に埋め込む LLE (Locally Linear Embedding) [4] を用いる。しかし LLE を用いるにあたり解決すべき問題が2つ生じてしまう。

1. LLE はデータ数 × データ数という行列の固有値分解を行うため、膨大なデータに対して LLE を適用することは現実的でない。
2. 一般にグラフスペクトル法は高次元空間から低次元空間への写像とその逆写像を得ることができない。

■問題1の解決法

[5] で提案された手法を用いてブロック量子化された行列に変形した後固有値分解を行う。

■問題2の解決法

LLE は高次元空間内のデータの局所線形性を保存したまま低次元空間に埋め込むので、低次元空間の局所共分散と対応する高次元空間における局所共分散は相互に対応していると考えられる。そこで本稿では高次元空間から低次元空間への写像、逆写像の代わりに低次元空間内での局所共分散を計算し、その計算結果を基に高次元空間での局所共分散を計算し KLT (2.2)、ビット割当 (2.3) を行う。以下に詳細な処理を記述する。

1. 低次元データに対して GMM を当てはめる。
2. 各コンポーネントの共分散、平均を計算する際に用いた各データ点の帰属率を基に高次元空間での各コンポーネントの共分散、平均を計算する。
3. 計算された共分散、平均を基に KLT (2.2)、ビット割当 (2.3) を行う。

2.2 KLT : Karhunen-Loeve Transform

KLT は、ベクトル x で構成されるデータセットの共分散行列に対する固有ベクトルからなる直交行列 P を用いてベクトル x を線形変換する。すなわち、

$$y = Px \quad (1)$$

逆変換は、

$$x = P^t y \quad (2)$$

† 広島市立大学大学院情報科学研究科
〒731-3194 広島県広島市安佐南区大塚東 3-4-1
Email: m.yamada@robotics.im.hiroshima-cu.ac.jp

となる。

本稿では、画像データを 8×8 ブロックへ分割した後 GMM をデータ分布に当てはめ、各クラスごとに固有値分解を行い変換行列 P_j ($j = 1, \dots, m$)、固有値 $\lambda_{j,k}$ ($j = 1, \dots, m, k = 1, \dots, n$) を得る。なお m, n はそれぞれ GMM のコンポーネント (クラス) 数、ブロックの要素数を表す。

2.3 ビット割当

本稿で用いるビット割当は、「クラス間ビット割当」、「クラス内ビット割当」から構成される。

■クラス間ビット割当

本稿で用いる変換符号化器は固定レート変換符号化器である。よって目標ビット数 (bpp: bits per pixel) を b とすると変換符号化器全体のビット数 b_{tot} は以下で表すことができる。

$$2^{b_{tot}} = 2^{nb} \quad (3)$$

次に、符号化器全体のビット数 b_{tot} をもとに、各クラスに割り当てられるビット数 b_j を以下の式に基づき計算する [2]。

$$2^{b_j} = 2^{b_{tot}} \frac{(c_j \Lambda_j)^{n/(n+2)}}{\sum_{l=1}^m (c_l \Lambda_l)^{n/(n+2)}} \quad (4)$$

$$\Lambda_j = \left[\prod_{k=1}^n \lambda_{j,k} \right]^{1/n} \quad j = 1, 2, \dots, m \quad (5)$$

ここで、 c_j はクラス j の混合率、 $\lambda_{j,k}$ はクラス j の k 番目の要素の固有値である。

■クラス内ビット割当

クラス間ビット割当により割り当てられた b_j をもとに、クラス内のブロックの各要素に対してビットを割り当てる [2]。

$$b_{j,k} = \frac{b_j}{n} + \frac{1}{2} \log_2 \frac{\lambda_{j,k}}{\left[\prod_{l=1}^n \lambda_{j,l} \right]^{1/n}} \quad j = 1, 2, \dots, m \quad (6)$$

ここで、 $b_{j,k}$ はクラス j の k 番目の要素に割り当てられたビット数である。

2.4 最小歪みブロック量子化

■量子化器の生成

各クラスの各要素に対するビット、固有値を用いて文献 [3] に記述されている n Gaussian Lloyd-Max scalar quantisers に基づき各クラスの各要素ごとに量子化器を構築する。

■新たな画像データへの圧縮処理

上記で訓練画像を基にした画像圧縮コーデックが作成された。以下では作成したコーデックを用いた新たな画像に対する圧縮処理について述べる。

圧縮データには各クラスで量子化した際の歪みが最小となるクラスのインデックスと量子化後のデータから構成される。圧縮データ作成の詳細を以下に示す。

1. 新たな画像のベクトル x_i を各クラスの変換行列 P_j 、平均値 μ_j 、固有値 λ_j を用いて標準偏差で正規化されたベクトル z_i^j に変換する。

$$z_i^j = \frac{P_j(x_i - \mu_j)}{\sqrt{\lambda_j}} \quad (7)$$

2. 正規化ベクトルを量子化器 Q_j を用いて量子化ベクトル \hat{z}_i^j に変換する。

3. 量子化ベクトル \hat{z}_i^j を復元ベクトル \hat{x}_i^j に変換する。

$$\hat{x}_i^j = P_j^t \sqrt{\lambda_j} \hat{z}_i^j + \mu_j \quad (8)$$

4. 歪みが最小となるクラス k_i を選択する

$$k_i = \arg \min_j d(x_i - \hat{x}_i^j) \quad (9)$$

3 実験と結果

訓練画像 17 枚を用いて、従来手法と本手法での画像圧縮コーデックを作成し (bpp=0.1~0.5, $m = 2, 4, 8$)、新たな画像 (訓練画像に含まれない) に対して圧縮、復元処理を行い PSNR (Peak Signal to Noise Ratio) を用いて復元誤差を評価した。以下にその結果の一部を記載する。

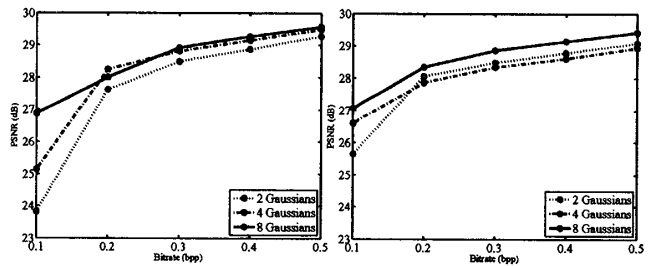


図1 左：従来手法 右：提案手法 (埋め込みビット数: 16)

結果から低ビットレートにおける圧縮性能が向上している。これはグラフスペクトル法を用いた長所であると言える。しかし、ビットレートが増加するに従い従来手法とほぼ同等の性能となっている。この結果については、今後 LLE と本稿での画像圧縮手法との相性も含めて議論していく必要があると考える。

4 まとめ

従来の KLT に基づく画像圧縮手法にグラフスペクトル法を組み込んだ新たな手法を提案した。本手法では複雑な高次元空間ではなく滑らかな低次元空間で GMM の当てはめを行うため、低ビットレートにおいても従来手法よりも良い結果を得ることができた。しかし低ビットレート以外では期待した結果を得ることはできなかった。今後の課題はより多くの実験と LLE に代わる画像圧縮に適したグラフスペクトル法の選別が挙げられる。

参考文献

- [1] K.K.Paliwal et al., Low-complexity GMM-based block quantisation of images using the discrete cosine transform, *Signal Processing: Image communication* 20, June 2005, pp. 435-446
- [2] A.D.Subramaniam et al., PDF Optimized Parametric Vector Quantization of Speech Line Spectral Frequency, *IEEE Trans. Speech Audio Processing* 11(2), March 2003, pp. 130-142
- [3] J.J.Y.Huang et al., Block Quantization of Correlated Gaussian Random Variables, *IEEE Trans. Commun. Syst. CS-11*, September 1963, pp. 289-296
- [4] L.K.Saul et al., Think Globally, Fit Locally: Unsupervised Learning of Nonlinear Manifolds, Technical Report MS CIS-02-18, 2002
- [5] K.Zhang et al., Block-Quantized Kernel Matrix for Fast Spectral Embedding, *International Conference on Machine Learning (ICML 2006)*, June 2006.