

領域分割に基づいたグラフカットによるグラフベースの カラリゼーション符号化の計算量削減手法

Image segmentation based graph-cut algorithm to reduce the computational cost of the color image coding using graph based colorization

安彦 魁人[†] 雨車 和憲[‡] 浜本 隆之[†] 半谷 精一郎[†]
Kaito Abiko Kazunori Uruma Takayuki Hamamoto Seiichirou Hangai

1 はじめに

カラリゼーションとは輝度画像と少量の色差情報を用いてカラー画像を復元する手法である。Levinの手法[1]は有名なカラリゼーション手法の一つであり、この手法では隣接画素の輝度値の類似度は色差値の類似度と強い相関があると仮定し、色差情報を最適化問題の最適解とすることによりカラー画像を復元する手法である。

近年、カラリゼーションを用いた画像符号化技術が研究されている[2, 3, 4]。この手法では、画像上のいくつかの画素の色差情報が選択・保存しておくことで符号化する。また、復号の際には保存していた色差情報からカラリゼーションを用いることによってカラー画像を復元する。いくつかの符号化アルゴリズムでは、JPEG2000よりも高い符号化効率を達成している。

文献[3, 4]ではグラフフーリエ変換に基づく符号化アルゴリズムが提案されている。このアルゴリズムでは輝度画像上のいくつかの画素がグラフの頂点として設定され、その画素の色差値がグラフ信号値として設定される。その後、色差値がグラフフーリエ変換に基づいてグラフスペクトルに変換され、そのスペクトルが圧縮され符号化される。また、復号の際にはグラフスペクトルを逆フーリエ変換に基づいて、頂点上の色差値が復元され、さらにカラリゼーションによってカラー画像が復元される。しかしこのアルゴリズムではグラフフーリエ変換にかかる計算時間が問題となっている。グラフフーリエ変換ではグラフラプラシアンと呼ばれる行列の固有値をいくつか求める必要がある。このグラフラプラシアンのサイズは頂点数に依存しており、高い符号化効率を実現するためには大きな行列でなければならないため、計算時間が膨大になる。

そこで本研究では計算時間を抑えるためにアルゴリズムの改良を行う。すなわち、高い符号化効率を維持したままに計算量を削減するために、画像の領域分割アルゴリズムに基づいたグラフカットを行う。そして分割された各セグメントごとに固有ベクトルを算出する。各セグメントのグラフラプラシアンのサイズは小さいので、短い時間で固有ベクトルを算出できる。その後、各固有ベクトルを画像全体の頂点に対応させる変換を行うことで、計算時間の削減を行う。本論文では、以上の処理過程を述べるとともに、12枚の画像を用いた実験により画質劣化を0.30dBに抑えて処理速度を16.8倍に向上できたことを示す。

2 グラフベースのカラリゼーション符号化

2.1 グラフフーリエ変換[5]

グラフラプラシアン $L \in R^{p \times p}$ はグラフの構造を表した行列である。ここで p はグラフの頂点数で

ある。いま、グラフラプラシアンの固有ベクトルを $v_i (i = 1, 2, \dots, p)$ とする。このとき固有ベクトル v_i はそれぞれに対応する固有値が小さい順に並んでいる。ここで、グラフフーリエ変換のためのグラフフーリエ基底行列 V は $V = [v_1, v_2, \dots, v_p]$ で定義される。グラフ信号処理では各頂点にグラフ信号値 $F_i (i = 1, 2, \dots, p)$ という値を持つ。また、グラフ信号値を順番に並べたベクトルを F とする。このとき、グラフフーリエ変換は以下の式で与えられる。

$$F = Vs, \quad (1)$$

ここで $s \in R^p$ はグラフスペクトルである。

2.2 カラリゼーション符号化

カラリゼーションでは輝度画像と色差情報を持ついくつかの画素から自然なカラー画像を得る手法である。近年、この技術を符号化に応用する手法が検討されており、カラリゼーション符号化と呼ばれている。この符号化方式ではいくつかの画素が符号化の際に選択され、その画素の色差情報を保存しておく。復号の際には保存しておいた輝度情報といくつかの色差情報からカラリゼーションを行うことによってカラー画像を復元する。

2.3 グラフベースのカラリゼーション符号化

グラフフーリエ変換を用いたカラリゼーション符号化が提案されている。このアルゴリズムでは、画像上のいくつかの画素をグラフの頂点として設定し、各頂点の物理的距離および輝度値からグラフラプラシアン L を与える。さらに L からカットオフグラフフーリエ基底行列 $\hat{V} \in R^{p \times c}$ を求める。ここで、カットオフグラフフーリエ基底行列 \hat{V} の列ベクトルはグラフフーリエ基底行列 V の最初の列ベクトルから c 番目までの列ベクトルに対応する。そしてグラフの頂点に対応する画素の色差情報をグラフ信号値 F として与える。

文献[3]のアルゴリズムでは符号化の際、カットオフグラフフーリエ変換 $\hat{s} = \hat{V}^T F$ に基づいてグラフ信号値 F がグラフスペクトルに変換される。復号の際には $F^* = \hat{V} \hat{s}$ に基づいて F^* が \hat{s} から得られる。このとき、 F^* は復元されたグラフ信号である。そしてカラリゼーションによって適切に復元されたカラー画像を得る。さらに改良されたアルゴリズムが文献[4]で提案されている。これらのアルゴリズムはJPEG2000以上の高い符号化効率を達成することができるが、固有ベクトルを計算する際の計算量が膨大となってしまう。

3 提案手法

本論文では従来手法[3, 4]の計算量を削減するためにグラフカットを用いた手法を提案する。

画像上のいくつか代表的な画素が選択された後、輝度画像を K 個の小領域に領域分割し、これらの小領域に従ってグラフをカットする。いま、グラフカットによっ

[†] 東京理科大学工学部電気工学科

[‡] 工学院大学情報学部コンピューター科学科

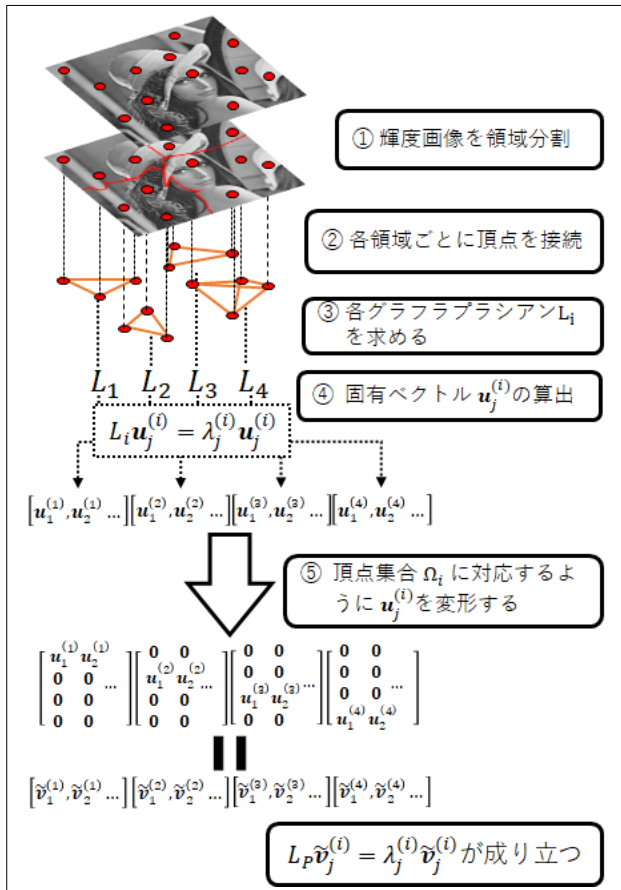


図1 提案手法の概略図

て接続の一部がカットされた p 個の頂点からなるグラフのグラフラプラシアンを L_P とする。また、 i 番目の小領域内の頂点集合を Ω_i とする。さらに、 Ω_i からなるグラフのグラフラプラシアンを L_i とし、その固有値及び固有ベクトルを $\lambda_j^{(i)}$ 及び $u_j^{(i)} \in R^{|\Omega_i|}$ ($j = 1, 2, \dots, |\Omega_i|$) とする。またこのとき固有値は小さい順に並んでいる。従って、 $L_i u_j^{(i)} = \lambda_j^{(i)} u_j^{(i)}$ と表せる。ここで、固有ベクトル $u_j^{(i)}$ に対してベクトル $\tilde{v}_j^{(i)} \in R^p$ を考える。 $\tilde{v}_j^{(i)}$ は Ω_i に対応する要素には $u_j^{(i)}$ の対応する要素が入り、他は0となるようなベクトルである。このとき以下が成り立つことが分かる。

$$L_P \tilde{v}_j^{(i)} = \lambda_j^{(i)} \tilde{v}_j^{(i)}. \quad (2)$$

すなわち小グラフのグラフフーリエ基底を求めることは全体のグラフフーリエ基底の非ゼロ部分の要素を求めることと等しい。また、この手法の概略図を図1に示す。

4 評価実験

図2は、分割数 K と処理時間の関係を調べた予備実験の結果である。Lena の画像を用いて実験を行う。また、 $K = 1$ が従来手法 [4] の結果である。結果から、16分割以降処理時間に大きな差がないことがわかる。したがって、以降 $K = 16$ と設定することとする。

次に提案手法と従来手法 [4]、及びブロック分割法の性能を比較する。ブロック分割法とは画像を16個の正方形の画像に分けそれぞれの画像を従来手法 [4] で符号化する手法である。12種類の 256×256 のサイズの標準画像を用いて符号化にかかる時間と復号画像の

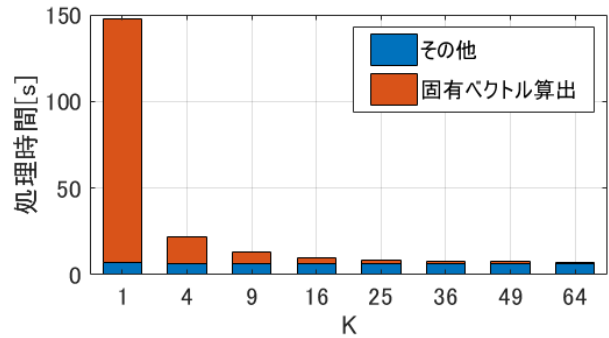
図2 分割数 K と処理時間の関係

表1 提案したアルゴリズムと従来手法の比較

		Lenna	Pepper	Milkdrop	平均
提案手法	time[s]	9.28	9.04	9.25	9.29
	PSNR[dB]	33.69	28.61	33.41	32.21
	SSIM	0.9903	0.9770	0.9902	0.9586
従来手法 [4]	time[s]	156.14	148.02	142.39	156.16
	PSNR[dB]	34.45	28.94	33.40	32.51
	SSIM	0.9919	0.9783	0.9895	0.9596
ブロック分割	time[s]	20.81	20.23	19.91	20.20
	PSNR[dB]	33.70	28.41	32.34	31.88
	SSIM	0.9903	0.9761	0.9869	0.9566

品質を調べた。表1に、処理時間と画質に関し、Lenna, Pepper, Milkdrop と12枚の画像に対する平均を示す。画質はPSNRとSSIM[6]を用いて比較する。結果より、提案手法は従来手法よりも16.8倍速い速度で符号化しており、PSNRは0.30dBのわずかな劣化であり、SSIMは同程度である。

5 おわりに

本論文は従来のカラリゼーション符号化アルゴリズムの高速アルゴリズムを提案した。画像の領域分割に基づいてグラフカットを行い、高い符号化効率を維持しながら計算時間を短縮した。数値結果ではPSNRの劣化を0.30dBに抑え、処理速度を16.8倍にすることができた。

参考文献

- [1] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," ACM Trans. Graph., vol. 23, no. 3, pp. 689-694, 2004.
- [2] S. Lee, S. Park, P. Oh, and M. Kang, "Colorizationbased compression using optimization," IEEE Trans. Image Process., vol. 22, no. 7, pp. 2627-2636, 2013.
- [3] K. Uruma, K. Saito, T. Takahashi, K. Konishi, T. Furukawa, "Representative pixels compression algorithm using graph signal processing for colorization-based image coding," In 2017 IEEE International Conference on Image Processing (ICIP), pp. 3255-3259, 2017.
- [4] K. Uruma, K. Konishi, T. Takahashi, T. Furukawa, "Colorization-based image coding using graph Fourier transform," Signal Processing: Image Communication, vol. 74, pp. 266-279, 2019.
- [5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644-1656, 2013.
- [6] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600-612, 2004.