

I-039

# スーパーピクセル分割を用いたカラリゼーション符号化に基づく深度画像符号化手法の提案

## Depth image coding algorithm based on the colorization based image coding using super-pixel segmentation

雨車和憲<sup>†</sup> 小西克巳<sup>‡</sup> 高橋智博<sup>†</sup> 古川利博<sup>†</sup>  
<sup>†</sup>東京理科大学工学部 <sup>‡</sup>工学院大学情報学部

Kazunori URUMA<sup>†</sup> Katsumi KONISHI<sup>‡</sup> Tomohiro TAKAHASHI<sup>†</sup> Toshihiro FURUKAWA<sup>†</sup>  
<sup>†</sup>Tokyo University of Science <sup>‡</sup>Kogakuin University

### あらまし

本研究では、カラー画像の高効率な符号化手法として近年注目されているカラリゼーション符号化手法を応用した、新しい深度画像符号化手法を提案する。深度画像について、符号化と復号化の両方においてこれに対応したカラー画像が使用できるものと仮定し、これを深度画像の符号化へ効果的に利用する方法を提案する。数値実験によって、JPEG2000と比較して提案手法が深度画像を効率的に圧縮できることを示す。

### 1 はじめに

近年、Kinectに代表されるような比較的安価な深度センサーが普及し、深度画像がさまざまな用途で利用されるようになった。これに伴い、深度画像の保存手法についても高い関心が集まっている。深度画像を通常の画像と同様JPEGやJPEG2000で保存した場合、視覚的な違和感の無さが優先され、人間の目では気づきにくいエッジなどに多く存在する高周波成分が失われる傾向にあるが、深度画像を利用する多くのアプリケーションでは、とくにエッジ周辺の深度値の正確さは重要である。そこで、より適切に深度画像を圧縮する手法についての研究が行われている[1, 2, 3]。文献[1]では、圧縮センシングの理論[4, 5]に基づいた方法が提案されており、JPEGやJPEG2000と比較して高い圧縮性能が示されている。また、文献[3]では、深度画像の復号化の際に、対応するカラー画像を用いて圧縮性能を向上させる手法が提案されている。

本研究では、より効率的な深度画像の圧縮を実現

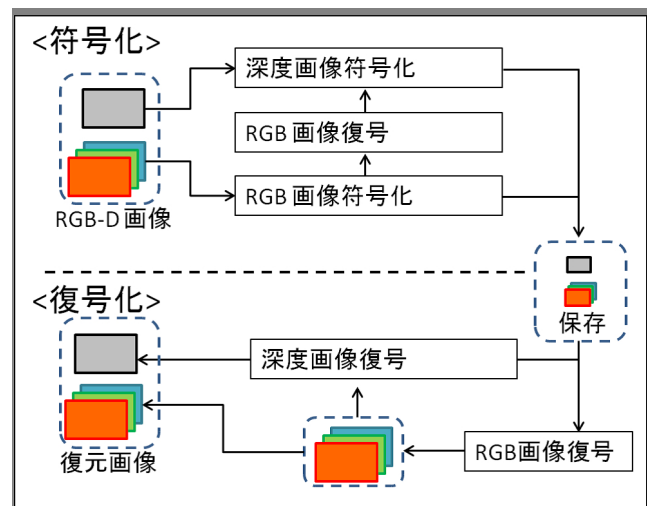


図 1: カラリゼーション符号化に基づく深度画像符号化手法の説明図

するために、カラリゼーション符号化と呼ばれるカラー画像圧縮手法に着目する[6, 7]。カラリゼーション符号化手法では、符号化の際、輝度成分は既存のJPEG圧縮などを用いて圧縮を行い、色差成分については適切に抽出された代表画素の色差情報のみを保存しておく。復号の際には、輝度画像と代表画素の色差情報をもとに、カラリゼーションと呼ばれるカラー画像復元手法[8, 9]によってカラー画像を復元する。このとき、抽出された代表画素は少ない保存容量で保存が可能であるため、高い圧縮性能が期待できる。先行研究ではJPEGやJPEG2000と比較して高い圧縮性能が示されている。

本研究では、符号化を行う深度画像に対応したカラー画像が、深度画像とともに保存されるものと仮

定し、カラリゼーション符号化に基づいた新しい深度画像符号化手法を提案する。図 1 に、提案するカラリゼーション符号化に基づいた深度画像符号化手法を示す。はじめに、符号化の際にはカラー画像を領域分割することによって求められる変換行列を用いて深度画像をスパースなベクトルとして表現し、これを保存する。復号の際には、符号化の際に求めた変換行列を再びカラー画像から計算し、保存しておいたスパースベクトルを用いて深度画像を復元する。このスパースなベクトルは少ない容量で保存可能なので、高い圧縮性能が実現できると期待できる。さらに、より高い圧縮性能を実現するために変換行列の各列ベクトルの特性に着目して、より適切にスパースベクトルを求める方法を提案する。

数値実験により提案手法が JPEG2000 と比較して高い圧縮性能を実現することを示す。

## 2 スパース最適化に基づくカラリゼーション符号化 [6]

ここでは、文献 [6] で Lee らによって提案されているカラリゼーション符号化手法について述べる。  $N$  を全画素数とし、  $\mathbf{u} \in R^N$  をベクトル表現された色差画像とする。また、  $M$  を正の定数として、行列  $C \in R^{N \times M}$  を輝度画像から一意に構成される変換行列と定義する。ただし、  $C$  の具体的な構成方法についてはあとで述べる。

はじめに符号化の際の処理について述べる。  $L$  ( $L < N, M$ ) を任意の正の定数として、深度画像を以下の問題の解として与えられるスパースベクトル  $\mathbf{x}^* \in R^M$  で表現する。

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & \|C\mathbf{x} - \mathbf{u}\|_2^2 \\ \text{Subject to} \quad & \|\mathbf{x}\|_0 \leq L, \end{aligned} \quad (1)$$

ただし、  $\|\mathbf{a}\|_0$  はベクトル  $\mathbf{a}$  の非ゼロの要素の数を表す。ここで、問題 (1) の解  $\mathbf{x}^*$  を色差情報として保存する。復号の際には、符号化時に構成した変換行列  $C$  を輝度画像から再び構成し、保存していた  $\mathbf{x}^*$  を用いて、次式より色差画像が復元される、

$$\mathbf{u}^* = C\mathbf{x}^*. \quad (2)$$

次に変換行列  $C$  の構成方法について述べる。輝度画像に対して、あらかじめ設定したパラメータを用いて領域分割を行い、該当する領域内を 1、そうでない領域を 0 で表す各領域に対するマスク画像を求める。分割数だけのマスク画像が得られるので、これ

らをベクトライズし、並べたものを行列  $C$  とする。ただし、1 度の領域分割の結果のみでは色差情報を正確に表現することはできないため、あらかじめ複数のパラメータを設定しておき、行列  $C$  の列が十分な数になるまで繰り返し領域分割を行う。このように、領域分割によって得られるマスク画像のスパースな結合によって色差情報を表現することにより、文献 [6] では高い圧縮率を達成している。しかし、値が 1 と 0 のマスク行列の線形結合のみで色差画像を表現すると、領域分割の結果が適切でない場合や、問題 (1) における定数  $L$  の設定が適切でない場合に、本来は存在しなかった疑似的なエッジが生じてしまう可能性がある。これを防ぐため、文献 [6] ではさらに、各マスク画像内における重心点からの距離に反比例した値を各要素値に掛けることによって、マスク画像内の値の変動を滑らかにする処理が行われており、これにより視覚的に良好な復元結果を与える手法となっている。

## 3 提案手法

本研究では、深度画像はこれに対応するカラー画像とともに保存、利用されるものと仮定し、前章で述べたカラリゼーション符号化を深度画像の符号化へ応用した新しい深度画像符号化手法を提案する。

### 3.1 カラリゼーション符号化に基づく深度画像符号化手法

$N$  を画像の全画素数、  $M, L$  をそれぞれ正の定数とし、  $\mathbf{d} \in R^N$  をベクトライズされた深度画像、  $A \in R^{N \times M}$  を既知のカラー画像から構成される変換行列と定義する。ただし、  $A \in R^{N \times M}$  の具体的な構成方法については後述する。

カラリゼーション符号化手法と同様、符号化の際には次のスパース最適化問題の解として与えられるスパースベクトル  $\mathbf{s}^*$  を求める、

$$\begin{aligned} \underset{\mathbf{s}}{\text{Minimize}} \quad & \|\mathbf{A}\mathbf{s} - \mathbf{d}\|_2^2 \\ \text{Subject to} \quad & \|\mathbf{s}\|_0 \leq L. \end{aligned} \quad (3)$$

一般に、この問題を解くことは困難であるので、本研究では orthogonal matching pursuit (OMP)[11, 12] アルゴリズムを用いて  $\mathbf{s}^*$  を計算し、保存を行う。このとき、  $\mathbf{s}^*$  を再構成するのに必要な情報は非ゼロの値のインデックスの集合  $\mathcal{S} \subset \mathcal{I}$  と、その要素の値  $s_i$ 、 ( $i \in \mathcal{S}$ ) である。ただし、  $\mathcal{I} = \{1, 2, \dots, M\}$  とする。そのため、保存に必要なデータ量  $Q$  は次式により与

えられる,

$$Q = M + Lq + 2q \text{ [bits]}, \quad (4)$$

ただし,  $q$  は各要素の値を保存する際に用いられる量子化ビット数であり, 右辺の第 3 項は量子化する際に用いる  $s^*$  の最大値および最小値を保存するために必要となるデータ量である.

復号の際には, 符号化時に構成した変換行列  $A$  をカラー画像から再び構成し, 保存していた  $s^*$  を用いて, 次式により深度画像  $d^*$  が復元される,

$$d^* = As^*. \quad (5)$$

次に, 既知のカラー画像から変換行列  $A$  を求める方法について述べる. 本研究では, 文献 [6] と同様に領域分割アルゴリズムを用いたアプローチによりカラー画像から変換行列  $A$  を求めることを提案する. 本研究では, 文献 [10] で提案されているスーパーピクセル分割アルゴリズムを用いる. このアルゴリズムは分割数を決定するパラメタ  $k$  と, 各分割領域の形に影響を与えるパラメタ  $l$  を入力とし, 適切に画像の領域分割を行うことができる. このとき, パラメタ  $l$  は大きい値になるにつれ四角い形の分割領域を構成する傾向にあり, この各分割領域のことをスーパーピクセルと呼ぶ.

スーパーピクセル分割のパラメタの集合を  $\Omega = \{(k_t, l_t) | t = 1, \dots, T\}$  と定義する. はじめに, パラメタ  $(k_1, l_1)$  を用いてカラー画像を領域分割し  $k_1$  個のスーパーピクセルを構成する. 各スーパーピクセル内に該当する領域内の画素値を 1, そうでない領域内の画素値を 0 で表すマスク画像を, 各スーパーピクセルごとに求める. この処理をすべてのパラメタについて行い,  $\sum_{t=1}^T k_t$  個のマスク画像を構成する. さいごに, これらのマスク画像をベクトライズし並べたものを変換行列  $A$  として定義する. 図 2 に提案する変換行列の構成方法を示す. 文献 [6] においては視覚的に良好な結果を与えるために各マスク画像内の値の変動を滑らかにする処理が加えられているが, カラー画像の符号化と異なり深度画像においては視覚的に良好であることよりもエッジの深度値がより正確であることの方が重要であると考え, 本研究ではこのような処理は行わない.

### 3.2 効率的な基底選択による深度画像符号化手法の性能改善

ここでは, 問題 (3) を解く際に行う行列  $A$  からの基底選択を効率的に行う手法を提案する. 1 度の領域

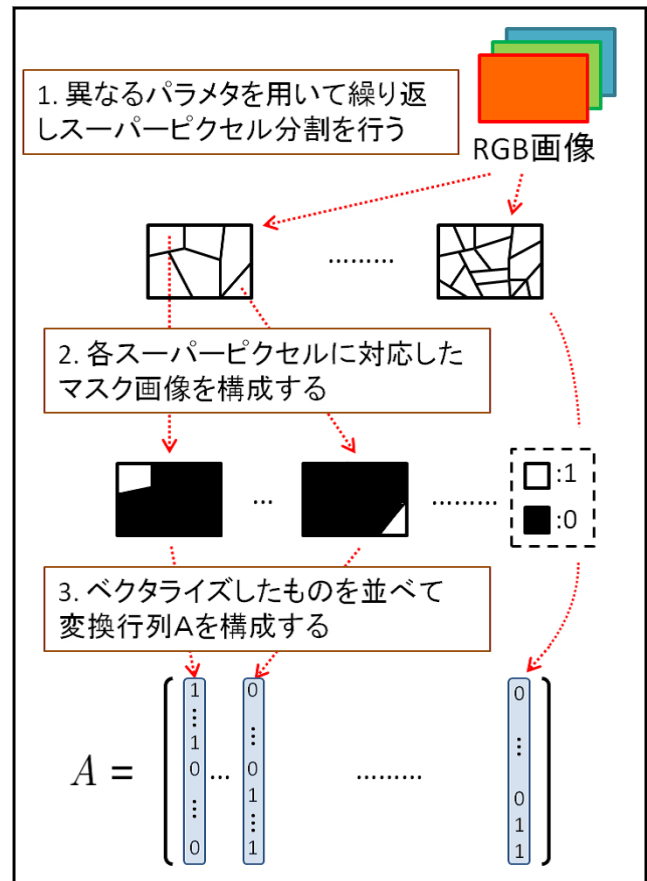


図 2: 変換行列作成方法の説明図

分割ごとにすべての画素がどこかのスーパーピクセルに属するため, 提案する変換行列の構成方法では, 各列ベクトルが  $k_t$  本ごとに互いに直交したベクトルの集合を構成する. 例えば, はじめの  $k_1$  本の列ベクトルに注目すると, これらは互いに直交しており, かつすべての要素に対していずれかのベクトルは値を持っているため, これらのみである程度正確に深度画像を表現することができる.

本研究では, 変換行列  $A$  から適切に基底を選択することと, 保存するデータ量をより少なく抑えることを目的として, 行列  $A$  から基底を選ぶ際にこれら  $k_1$  本の列ベクトルは必ず選択されるものとあらかじめ決めておくことを提案する. つまり, 本研究では, 問題 (3) を解く際に  $1, 2, \dots, k_1 \in S$  とした状態から OMP を行うことを提案する.

ここで, データの保存方法についても修正を加える. まず, はじめの  $k_1$  個のインデックスは非ゼロだと分かっているので, 保存に必要なデータ量は式 (4) から次式へと修正される,

$$Q = M - k_1 + Lq + 2q \text{ [bits]}. \quad (6)$$

また、行列  $A$  のはじめの  $k_1$  個の列ベクトルは互いに直交しており、これらのみで、ある程度正確に深度画像を表現することができることから、OMP の結果得られる  $s^*$  のはじめの  $k_1$  個の要素の値はそれ以外と比べ重要であると考えられる。そこで、本研究でははじめの  $k_1$  個の要素に対する量子化ビット数を、それ以外の要素に対する量子化ビット数より大きく設定することで、より効率的にデータを保存する手法を提案する。はじめの  $k_1$  個の要素に対する量子化ビット数を  $q_1$ 、それ以外の要素に対する量子化ビット数を  $q_2$  として、保存の際に必要なデータ量  $Q$  を式 (6) から以下の式へと再び修正する、

$$Q = M - k_1 + q_1 k_1 + q_2 (L - k_1) + 2q_1 + 2q_2 \text{ [bits]}. \quad (7)$$

#### 4 数値実験

本研究では、Middlebury データセット [13, 14] を使用し、JPEG2000 との比較を行った。実験に用いたテスト画像を図 3 に示す。提案手法において用いたパラメータは  $q_1 = 8$ ,  $q_2 = 6$ ,  $(k_2, l_2)$  から  $(k_{10}, l_{10})$  については  $k = \{50, 100, 200\}$  および  $l = \{1, 5, 10\}$  のすべての組み合わせを用いた。また、 $l_1 = 10$ ,  $L = 200 + k_1$  とし、 $k_1$  の値については  $k_1 = \{200, 300, 400, 500, 800, 1200\}$  を用いたときの実験をそれぞれ行った。提案手法の結果は RGB 画像の領域分割の結果から影響を受けるため、RGB 画像を peak signal to noise ratio (PSNR) の値が 25 dB または 35 dB となるように JPEG2000 により圧縮を行ったものと、圧縮を行わなかった場合の 3 通りについて実験を行った。また、画像の評価については、次式で定義される mean absolute difference (MAD) の値を用いた、

$$\text{MAD} = \frac{1}{N} \sum_{i=1}^N |d_i^* - d_i|, \quad (8)$$

ただし、 $d_i^*$  と  $d_i$  はそれぞれ復号結果の深度画像  $d^*$  とオリジナルの深度画像  $d$  の  $i$  番目の値であり、 $N$  は全画素数を表す。数値実験の結果として、横軸に深度画像の保存に要したデータ量、縦軸に MAD の値を持つグラフを図 4 に示す。RGB 画像の PSNR が 35 dB および非圧縮の場合に、提案手法が高い圧縮性能を示した。とくに高圧縮率の際には、より優位性が示される結果となった。また、RGB 画像の PSNR が 25 dB の場合には、領域分割が適切に行われなかったために変換行列  $A$  が適切に構成されなかったことが原因で、効率的に符号化が行われなかったものと

考えられる。図 5 に復元結果の画像を示す。提案手法において利用した RGB 画像の PSNR は 35 とした。視覚的にどちらがよいかの判断は難しいと考えられるが、エッジなどの情報は提案手法の結果の方がより正確に復元されていることがわかる。提案手法は高圧縮率の場合においても各オブジェクトの輪郭の特徴が適切に保存されていることがわかる。

#### 5 むすび

本研究では、カラリゼーション符号化を深度画像の圧縮手法へ適用する手法を提案した。カラー画像に対しあらかじめ設定しておいたパラメータを用いて複数回領域分割を行い、符号化と復号化の両方で同じ変換行列を構成し、これを用いて深度画像を適切に圧縮することにより高い圧縮性能を実現する手法を提案した。さらに、符号化の時に保存を行う基底ベクトルの選択において、基底行列の各列ベクトルの性質に着目して、より高い性能を実現する手法について提案した。4 枚の画像を用いて数値実験を行い、JPEG2000 と比較して高い圧縮性能が示された。

#### 謝辞

本研究は JSPS 科研費 26・6546 の助成を受けたものです。

#### 参考文献

- [1] M. Sarkis and K. Diepold, "Depth map compression via compressed sensing," in Proc. IEEE Int. conf. Image Process. (ICIP), pp. 737-740, 2009.
- [2] T. Wang, H. Bai, M. Liu, C. Lin and Y. Zhao, "Depth map coding based on adaptive block compressive sensing," IEEE China Summit and Int. Conf. Signal and Inf. Process. (ChinaSIP), pp.492-495, 2015.
- [3] E. C. Forster, T. Lowe, S. Wenger, and M. Magnor, "RGB-Guided Depth Map Compression via Compressed Sensing and Sparse Coding," Picture Coding Symp. (PCS), pp. 1-4, 2015.
- [4] E. J. Candes, J. Romberg, and T. Tao, "Stable signal recovery for incomplete and inaccurate measurements," Commun. Pure Appl. Math., vol. 59, pp. 1207-1223, Aug. 2006.

- [5] D. L. Donoho, "Compressed sensing," IEEE Trans. Inf. Theory, vol. 52, pp. 1289-1306, Apr. 2006.
- [6] S. Lee, SW. Park, P. Oh, and MG. Kang, "Colorization-based compression using optimization," IEEE Trans. Image Process., vol. 22, no.7, pp. 2627-2636, 2013.
- [7] K. Uruma, K. Konishi, T. Takahashi and T. Furukawa, "Fast colorization based image coding algorithm using multiple resolution images," EURASIP Journal on Image and Video Process., 2016:7, pp.1-15, 2016.
- [8] A. Levin, D. Lischinski and Y. Weiss, "Colorization using optimization," ACM Trans. Graphics, Vol. 23, No. 3, pp. 689-694, 2004.
- [9] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," IEEE Trans. Image Process., Vol. 15, No. 5, pp. 1120-1129, 2006.
- [10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," IEEE Trans. PAMI, vol. 34, no.11, 2012.
- [11] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," SIAM J. Sci. Comput., vol. 20, no. 1, pp. 33-61, 1998.
- [12] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Inf. Theory, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.
- [13] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," In Proc. CVPR, 2007.
- [14] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," In Proc. CVPR, 2007.



図 3: テスト画像. 上から Art, Laundry, Dolls, Moebius(左: 深度画像, 右: カラー画像). サイズはすべて  $512 \times 512$ .

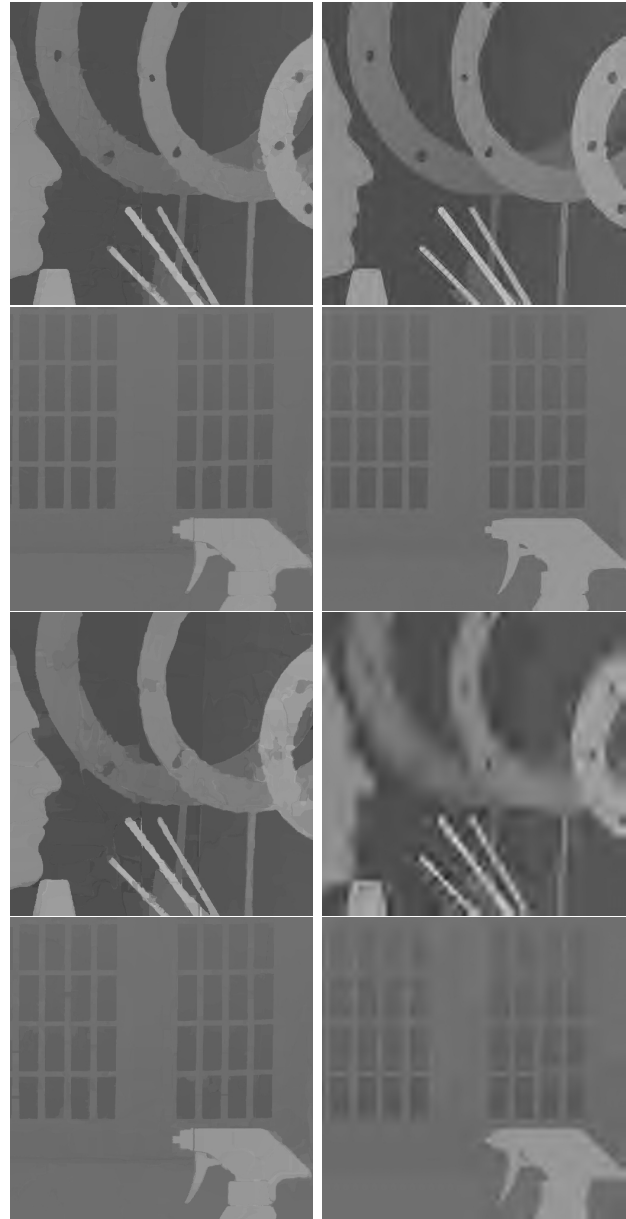
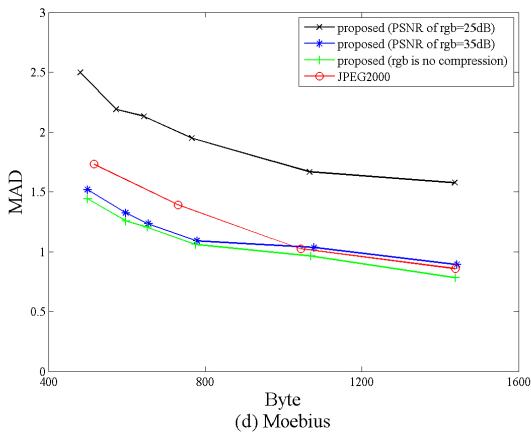
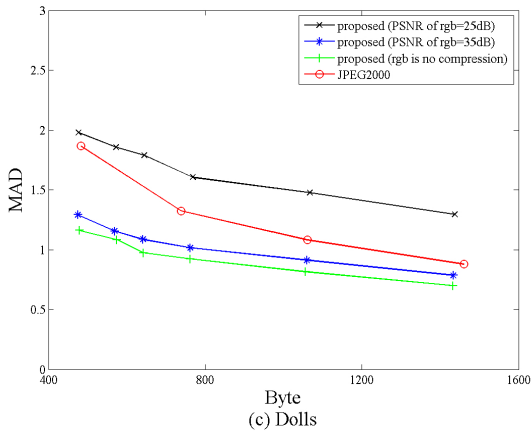
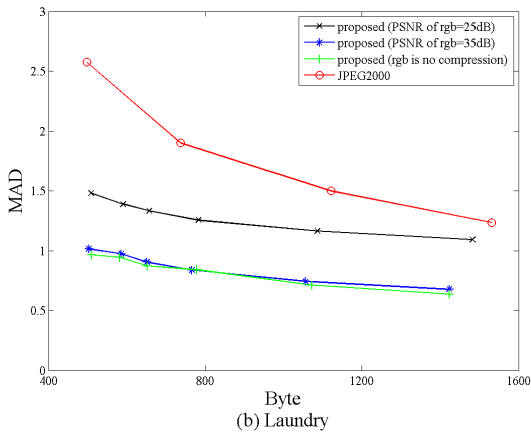
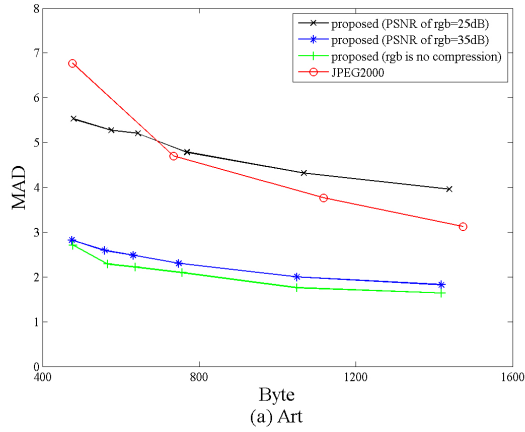


図5: 数値実験結果画像。左が提案手法による復元画像(上から, MAD=1.82, 0.68, 2.82, 1.01, 保存容量=1419, 1424, 475, 502[byte]), 右がJPEG2000による復元結果画像(上から, MAD=3.12, 1.23, 6.77, 2.49, 保存容量=1474, 1531, 476, 528[byte]).

図4: 数値実験結果. 縦軸がMAD, 横軸が保存容量[byte].