Uruma et al. EURASIP Journal on Image and Video Processing (2016) 2016:7
DOI 10.1186/s13640-016-0107-8

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Fast colorization based image coding algorithm using multiple resolution images

Kazunori Uruma[1*†], Katsumi Konishi[2†], Tomohiro Takahashi[1] and Toshihiro Furukawa[1]

Abstract

This paper proposes a representative pixel (RP) extraction algorithm and chrominance image recovery algorithm for the colorization-based digital image coding. The colorization-based coding methods reduce the color information of an image and achieve higher compression ratio than JPEG coding; however, they take much more computing time. In order to achieve low computational cost, this paper proposes the algorithm using the set of multiple-resolution images obtained by colorization error minimizing method. This algorithm extracts RPs from each resolution image and colorizes each resolution image utilizing a lower resolution color image, which leads to the reduction of the number of RPs and computing time. Numerical examples show that the proposed algorithm extracts the RPs and recovers the color image fast and effectively.

Keywords: Image colorization, Image coding, Representative pixel, Multiple-resolution images

1 Introduction

The image colorization technique can recover a full-color image from a luminance image and several representative pixels (RPs) which have the chrominance values and their positions [1–4]. Levin et al. proposes a well known colorization algorithm, which achieves a high colorization performance using appropriately given RPs. However, this algorithm requires high computing cost and fails to colorize a whole image if only a few RPs are given. The authors have proposed the colorization algorithm, which can restore a color image from only a few RPs and takes a low computational cost [4].

The development of the colorization technique has led to the generation of a new color image coding method called colorization-based color image coding [5–12]. Because the luminance image can be compressed by standard coding methods such as JPEG coding and because the chrominance information is represented by a small number of RPs, the colorization-based image coding can compress full-color images more than these methods. The colorization-based image coding algorithms proposed in [10, 12] have achieved higher coding performance than JPEG2000, which is a highly optimized standard using many entropy coding techniques together. Therefore, the colorization-based image coding has potential to become standard image coding for the next generation. The performance of the colorization based coding is evaluated based on the number of RPs and the quality of a restore image, and therefore, the objective of this paper is to propose a new coding method which requires less RPs and gives more accurate chrominance information. In order to reduce the amount of information to represent the RPs, the method [5] assumes that all pixels in a line segment have only one color and proposes the algorithm where the RPs are described as a set of color line segments. In [9], the algorithm finds a set of two RPs which give the similar effect for colorization result, and the redundant RPs are deleted. In [12], under the assumption that the chrominance image is given as a sparse linear combination of the basis constructed from the luminance image, the algorithm achieves a high compression performance using the theory of compressed sensing [13, 14]. However, these algorithms require a high computing time depending on the image size. To reduce computing time, the method [5] proposes an algorithm to extract RPs from a single low-resolution image. This paper takes the same approach to reduce the computational costs and

*Correspondence: uru-kaz@ms.kagu.tus.ac.jp
†Equal contributors
[1]Tokyo University of Science, 162-8677 Tokyo, Japan
Full list of author information is available at the end of the article

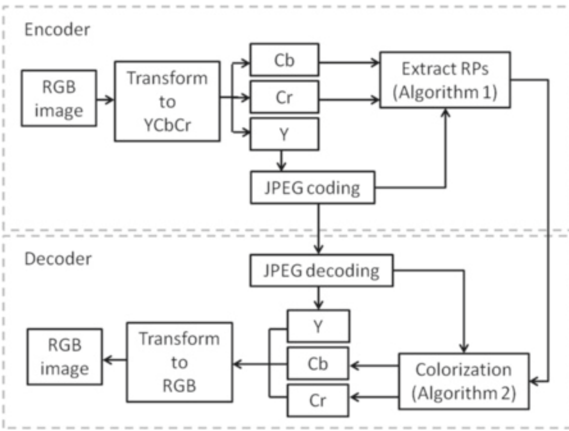

Fig. 1 Outline of the proposed colorization-based image coding

provides an RPs extraction algorithm using multiple low-resolution images. Though a low resolution image used in [5] is obtained by a simple downsampling, this paper gives low resolution images which are optimized such that a good color image is recovered in the colorization phase.

This paper proposes a new colorization-based image coding algorithm consisting of the extraction RPs phase (coding) and the colorization phase (decoding). In the phase of extracting RPs, the algorithm extracts RPs using error feedback between the original image and the colorized image similar to [15]. In order to achieve a low computational cost and to reduce the amount of information to represent RPs, the algorithm extracts RPs from a set of multiple resolution images obtained by multiple downsampling and colorization error minimizing. In the phase of colorization, the colorization algorithm colorizes multiple-resolution images. Performance of the colorization algorithm affects the performance of the colorization-based image coding, that is, the colorization using a few color pixels leads to a high coding performance. To propose a fast and precise coding algorithm, this paper modifies the colorization algorithm proposed in [4] and introduces a multiple resolution scheme to reduce the computational cost. Figures 1 and 2 show the outline and concept of the proposed algorithm, respectively. The major contribution of this paper is to propose a colorization-based image coding algorithm which requires less computational time and achieves a high coding performance.

This paper is organized as follows. In Section 2.2, we give a colorization algorithm based on the algorithm proposed in [4]. Section 2.3 proposes the RPs extraction algorithm which uses recovery error feedback, and it is applied to multiple resolution images. In Section 2.4, a multi-resolution color image recovery algorithm is proposed. We consider the implementation of saving images with RPs in Section 2.5. Numerical

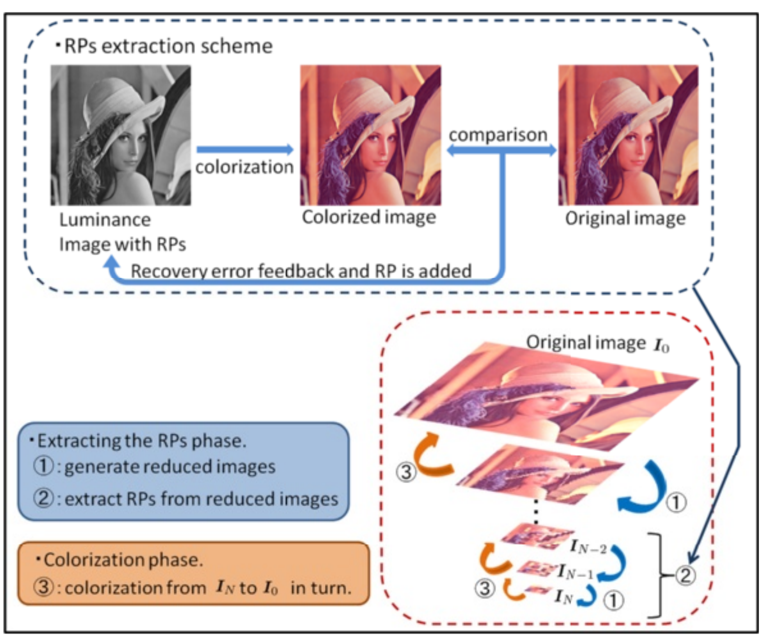

Fig. 2 Concept of the proposed algorithms

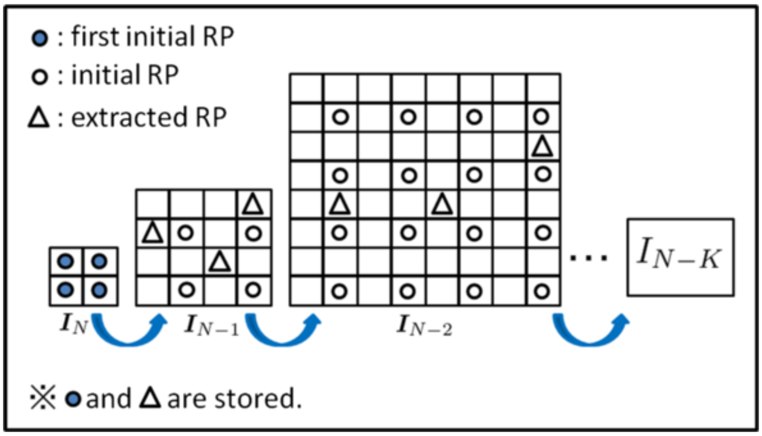

Fig. 3 Outline of the proposed RP extraction algorithm using multiple resolution images

examples show that the proposed algorithm extracts the RPs and recovers the color image fast and effectively in Section 3.

2 Main works

2.1 Notation

We provide here a brief summary of the notations used throughout the paper. $(A)_{i,j}$ and $(\boldsymbol{a})_i$ denote the (i,j)-element of a matrix A and the ith element of a vector $\boldsymbol{a}$. $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the l_2 norm of a vector and the Frobenius norm of a matrix, respectively. $\mathbf{0}_k \in R^k$ is denoted as a k-dimension zero vector. We use $\mathrm{diag}(A_1,\ldots,A_m)$ to denote a block diagonal matrix consisting of $A_1, \ldots, A_m$ and $|S|$ to denote the number of elements in a set S. We denote the ceiling function and the floor function by $\lceil a \rceil$ and $\lfloor a \rfloor$, respectively.

2.2 Colorization algorithm

In order to provide a colorization-based image coding, this paper applies the colorization algorithm proposed in [4] because it can recover a full color image from a luminance image using only a few color pixels. The algorithm restores chrominance images C_b and C_r independently to obtain a color image, and this subsection gives a chrominance image restoration method.

Let vectors $\boldsymbol{x} \in \boldsymbol{R}^{mn}$ and $\boldsymbol{y} \in \boldsymbol{R}^{mn}$ denote a desired chrominance image and a rasterized-given luminance image, respectively, where m and n denote height and width of the image. Define $U \in \boldsymbol{R}^{(m-1)\times m}$, $V \in \boldsymbol{R}^{m(n-1)\times mn}$, $\bar{U} \in \boldsymbol{R}^{(m-1)n\times mn}$ and $D \in \boldsymbol{R}^{(2mn-m-n)\times mn}$ as

$$U_{i,j} = \begin{cases} 1, & \text{if } i = j \\ -1, & \text{if } i+1 = j \\ 0, & \text{otherwise} \end{cases},$$

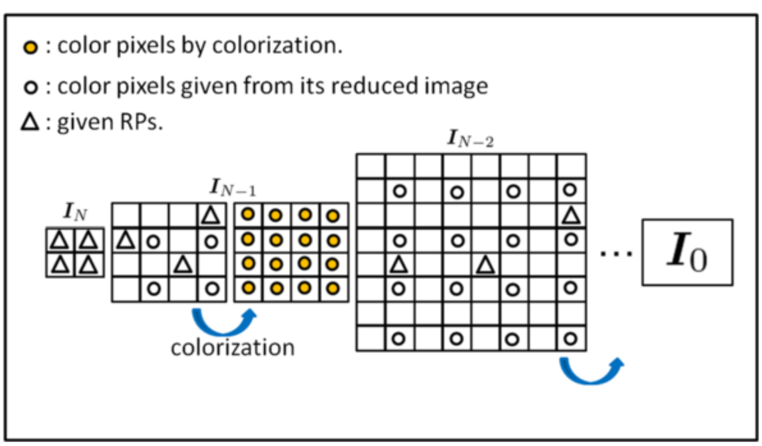

Fig. 4 Outline of the proposed chrominance image recovery algorithm using multiple resolution images

Table 1 Correspondence table (bit unit is 3)

Length	Sign
9	111001
8	111000
7	110
...	...
2	010
1	001
0	000

$$V_{i,j} = \begin{cases} 1, & \text{if } i = j \\ -1, & \text{if } i + M = j \\ 0, & \text{otherwise} \end{cases},$$

$$\bar{U} = \mathrm{diag}(U, \ldots, U),$$

and

$$D = \left[\bar{U}^T \; V^T\right]^T, \tag{1}$$

respectively. The matrices $\bar{U}$ and V denote vertical and horizontal difference operators. Then $D\boldsymbol{x}$ denotes the differences between the neighbor pixels of a whole image. Let us consider the following ℓ_2 norm minimizing colorization problem,

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{Minimize}} && \|D\boldsymbol{x}\|_2^2 \\ &\text{Subject to} && (\boldsymbol{x})_i = c_i, \forall i \in \mathcal{C}, \end{aligned} \tag{2}$$

where c_i is a constant corresponding to a given color value, and the index set $\mathcal{C}$ denotes a given set of vector indices of given color pixels. This problem recovers a color image by minimizing the sum of differences between neighbor pixels. In order to colorize an image more appropriately, [4] assumes that the differences between neighbor pixels of a chrominance image have relationship with those of a color image, which is defined by the following equality,

$$(D\boldsymbol{x})_i = \alpha f((D\boldsymbol{y})_i), \tag{3}$$

where α is an unknown constant, and $f : R \to R$ is a given function. Then, we obtain the following problem from (2) under the assumption (3),

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{Minimize}} && \|FD\boldsymbol{x}\|_2^2 \\ &\text{Subject to} && (\boldsymbol{x})_i = c_i, \forall i \in \mathcal{C}, \end{aligned} \tag{4}$$

where F is a diagonal matrix whose ith diagonal element is $1/f((D\boldsymbol{y})_i)$. The details are described in [4]. While the problem (4) is a convex quadratic programming and can

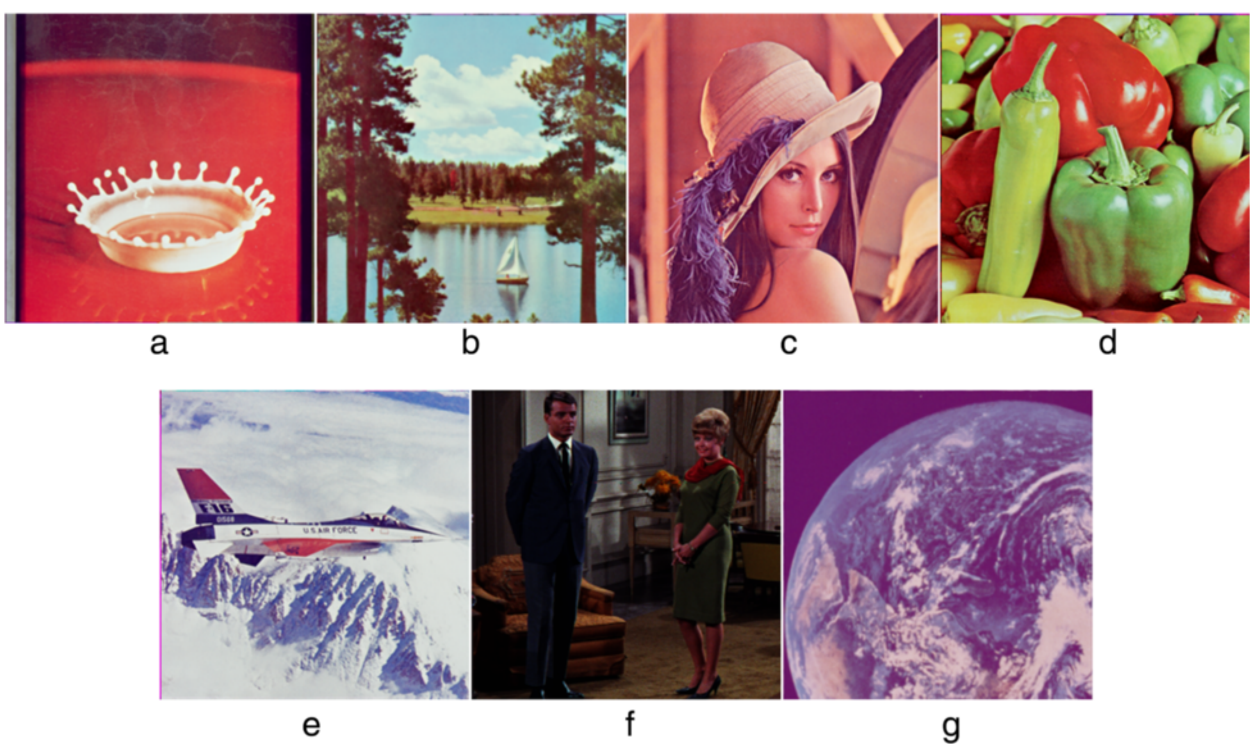

Fig. 5 Test images (256 × 256): **a** milkdrop, **b** boat, **c** lenna, **d** pepper, **e** airplane, **f** couple, and **g** earth

be solved exactly, [4] applies Lagrangian relaxation to reduce the computational cost. This paper provides an exact analytical solution of (4) which can be implemented simply.

Let us define $H \in R^{(2mn-m-n)\times mn}$ by FD. Then, we define $H^* \in R^{(2mn-m-n)\times(mn-p)}$ and $\acute{H}^* \in R^{(2mn-m-n)\times p}$ as submatrices of H composed of columns corresponding to unknown color pixels and given color pixels, respectively. Let $\boldsymbol{x^*} \in R^{mn-p}$ and $\acute{\boldsymbol{x}}^{\boldsymbol{*}} \in R^p$ denote unknown color vectors and given color vectors, respectively, where p denotes the number of given color pixels. Then the objective function of (4) is rewritten as follows,

$$FD\boldsymbol{x} = H\boldsymbol{x} = H^*\boldsymbol{x^*} + \acute{H}^*\acute{\boldsymbol{x}}^{\boldsymbol{*}}. \tag{5}$$

Because all entries of $\acute{\boldsymbol{x}}^{\boldsymbol{*}}$ are given by c_i, that is, they satisfy the constraint of (4), the problem (4) is equal to the following nonconstrained convex quadratic programming,

$$\underset{\boldsymbol{x^*}}{\text{Minimize}} \qquad \|H^*\boldsymbol{x^*} + \acute{H}^*\acute{\boldsymbol{c}}^{\boldsymbol{*}}\|_2^2, \tag{6}$$

where vector $\acute{\boldsymbol{c}}^{\boldsymbol{*}} \in R^p$ is a given vector consisting of c_i. If ${H^*}^T H^*$ is nonsingular, we obtain the solution of (6) as follows,

$$\boldsymbol{x^*} = -\left({H^*}^T H^*\right)^{-1} {H^*}^T \acute{H}^* \acute{\boldsymbol{c}}^{\boldsymbol{*}}. \tag{7}$$

Now we move onto discuss the singularity of ${H^*}^T H^*$. For an image $\boldsymbol{a} = [a_1, a_2, \ldots, a_{mn}]^T \in \boldsymbol{R}^{mn}$, the following lemma is provided.

Lemma 1. *It holds that* $H\mathbf{a} = \mathbf{0}_{2mn-m-n}$ *if and only if* $a_1 = a_2 = \cdots = a_{mn}$.

Proof. Because F is an invertible matrix, $H\boldsymbol{a} = \mathbf{0}_{2mn-m-n}$ if and only if $D\boldsymbol{a} = \mathbf{0}_{2mn-m-n}$. Since D is the difference operator defined by (1), $D\boldsymbol{a}$ is a zero vector if and only if all pixels have the same values, that is, $a_1 = a_2 = \cdots = a_{mn}$. □

Let $\boldsymbol{a}^{(i)} \in R^{mn-1}$ denotes a subvector of $\boldsymbol{a}$ and be composed by deleting an ith element of $\boldsymbol{a}$, and $H^{(i)} \in R^{(2mn-m-n)\times(mn-1)}$ is denoted as a submatrix of H and is composed by deleting the ith column of H. Then the Lemma 1 leads to the following lemma:

Lemma 2. *For all* $i \in \{1, 2, \ldots, mn\}$, *it holds that* $H^{(i)}\boldsymbol{a}^{(i)} = \mathbf{0}_{2mn-m-n}$ *if and only if* $\boldsymbol{a}^{(i)} = \mathbf{0}_{mn-1}$.

Proof. We have that

$$H^{(i)}\boldsymbol{a}^{(i)} = H\boldsymbol{a} - H\acute{\boldsymbol{a}} = H(\boldsymbol{a} - \acute{\boldsymbol{a}}), \tag{8}$$

where $\acute{\boldsymbol{a}} = [0, \ldots, 0, a_i, 0, \ldots, 0]^T \in \boldsymbol{R}^{mn}$. Therefore, we obtain followings from Lemma 1,

$$\begin{aligned} H^{(i)}\boldsymbol{a}^{(i)} = \mathbf{0}_{2mn-m-n} &\Leftrightarrow H(\boldsymbol{a} - \acute{\boldsymbol{a}}) = \mathbf{0}_{2mn-m-n} \\ &\Leftrightarrow a_1 = a_2 = \ldots = a_i - a_i = \ldots = a_{mn} \\ &\Leftrightarrow a_1 = a_2 = \cdots = 0 = \ldots = a_{mn} \\ &\Leftrightarrow \boldsymbol{a}^{(i)} = \mathbf{0}_{mn-1}. \end{aligned} \tag{9}$$

□

Theorem 1. *${H^*}^T H^*$ is a nonsingular matrix*

Proof. Lemma 2 guarantees that the subspace of the column space of H is linear independent, that is, the column space of H^* is linear independent. □

From this theorem, there exists the inverse of ${H^*}^T H^*$, and therefore we can obtain the exact solution of (4) by (7).

2.3 RPs extraction algorithm

This subsection proposes an algorithm to extract the RPs from an original image utilizing the error feedback between the original image and the colorized image. Let $Y \in \boldsymbol{R}^{m\times n}$, $Cb \in \boldsymbol{R}^{m\times n}$ and $Cr \in \boldsymbol{R}^{m\times n}$ denote a given luminance image and two given chrominance images, respectively. First we provide the following RPs extraction scheme for a given color image, its luminance image, a given initial set of RPs and a small constant $\varepsilon > 0$,

Step 1. Recover two chrominance images $\bar{Cb}$ and $\bar{Cr}$ using (7) with the luminance image and current RPs.
Step 2. Calculate the error between the original image and the colorized image obtained in Step 1 at each pixel as follows,

$$(E)_{i,j} = |(Cb - \bar{Cb})_{i,j}| + |(Cr - \bar{Cr})_{i,j}|. \tag{10}$$

Step 3. Terminate if $(E)_{i,j} \le \varepsilon$ for all (i,j), else go to Step 4.
Step 4. Add the pixel with the largest error $E_{i,j}$ to the set of RPs.
Step 5. Terminate if the number of iterations is larger than a given threshold, else go to Step 1.

Since this scheme extracts one pixel in one iteration, some pixels are selected appropriately as RPs by repeating Step 1–5. Although this scheme can steadily reduce the recovery error, it requires a lot of iterations to obtain an adequate amount of RPs and takes a lot of computing time for a large size image. In order to reduce the computing cost, this paper proposes an RP extraction algorithm using multiple resolution images.

Table 2 Numerical results for several (N, K)s

Milkdrop	N	N-K 0	1	2	3	Airplane	N	N-K 0	1	2	3
PSNR [dB]	4	38.0759	37.5711	37.1943	34.3947	PSNR [dB]	4	38.2035	33.2393	32.3945	32.0054
	5	37.6920	37.2186	36.8684	34.2402		5	38.1185	33.2027	32.3641	31.9780
	6	37.6314	37.1643	36.8181	34.2058		6	37.7937	33.1094	32.2888	31.9157
	7	37.6244	37.1576	36.8116	34.2006		7	37.7754	33.1061	32.2874	31.9159
Encoding Time [s]	4	14.1671	2.3608	0.6417	0.2239	Encoding Time [s]	4	14.6127	2.3328	0.6553	0.2307
	5	13.9903	2.4329	0.7173	0.2841		5	14.3815	2.3910	0.7077	0.2871
	6	13.9673	2.4428	0.7311	0.3021		6	14.6817	2.4039	0.7209	0.3002
	7	14.0246	2.4431	0.7268	0.3098		7	14.7841	2.4064	0.7275	0.3042
Decoding Time [s]	4	0.2144	0.2189	0.2122	0.2170	Decoding Time [s]	4	0.2099	0.2111	0.2033	0.1969
	5	0.2137	0.1979	0.2156	0.1996		5	0.2240	0.1994	0.2011	0.1988
	6	0.2382	0.2101	0.2075	0.1974		6	0.2193	0.2040	0.2087	0.2146
	7	0.2242	0.2021	0.2336	0.2136		7	0.2181	0.1986	0.2042	0.2231
Volume [byte]	4	1407	1146	913	702	Volume [byte]	4	1411	1148	915	706
	5	1193	933	699	488		5	1197	935	702	492
	6	1195	933	700	489		6	1154	892	659	449
	7	1193	932	699	488		7	1149	886	653	444
Boat	N	N-K 0	1	2	3	Couple	N	N-K 0	1	2	3
PSNR [dB]	4	34.2850	33.9851	33.6053	32.5525	PSNR [dB]	4	38.5062	38.2618	37.9755	37.0003
	5	34.0394	33.7643	33.4117	32.4187		5	38.3833	38.1462	37.8793	36.9063
	6	34.0133	33.7336	33.3841	32.3973		6	38.3118	38.0789	37.8241	36.8307
	7	34.0132	33.7335	33.3840	32.3969		7	38.3085	38.0757	37.8214	36.8280
Encoding Time [s]	4	14.117	2.3290	0.6535	0.2281	Encoding Time [s]	4	14.7000	2.3234	0.6541	0.2365
	5	14.1622	2.4299	0.7107	0.2882		5	14.3138	2.3749	0.7144	0.2868
	6	14.2034	2.4091	0.7348	0.3129		6	14.3077	2.3927	0.7312	0.3089
	7	14.0931	2.4581	0.7512	0.3148		7	14.4525	2.4048	0.7412	0.3151
Decoding Time [s]	4	0.2119	0.2098	0.2359	0.1954	Decoding Time [s]	4	0.2105	0.1967	0.1976	0.2081
	5	0.2111	0.2058	0.2159	0.2064		5	0.2072	0.1983	0.1980	0.2156
	6	0.2162	0.2037	0.2016	0.1980		6	0.2075	0.2159	0.2026	0.2246
	7	0.2116	0.2069	0.2099	0.2536		7	0.2068	0.2166	0.2120	0.2006
Volume [byte]	4	1413	1150	912	702	Volume [byte]	4	1414	1154	917	705
	5	1198	936	698	487		5	1200	940	703	491
	6	1195	933	695	484		6	1194	934	697	485
	7	1198	936	698	487		7	1191	931	694	482
Lenna	N	N-K 0	1	2	3	Earth	N	N-K 0	1	2	3
PSNR [dB]	4	36.3927	36.2937	36.0081	35.3116	PSNR [dB]	4	38.5740	38.4291	38.1530	37.4579
	5	36.1112	36.0174	35.7481	35.1009		5	38.1631	38.0269	37.7843	37.0972
	6	36.0689	35.9747	35.7079	35.0432		6	38.0894	37.9473	37.7087	37.0341
	7	36.0716	35.9774	35.7105	35.0497		7	38.0962	37.9539	37.7153	37.0421

Table 2 Numerical results for several (N, K)s *(Continued)*

Encoding Time [s]	4	14.0836	2.2984	0.6657	0.2247	Encoding Time [s]	4	14.5969	2.2922	0.6481	0.2284
	5	14.1050	2.3610	0.7059	0.2838		5	14.2419	2.3374	0.7004	0.2842
	6	14.4354	2.3993	0.7309	0.3194		6	14.3422	2.3709	0.7293	0.3078
	7	14.5874	2.3534	0.7317	0.3150		7	14.1882	2.3787	0.7252	0.3131
Decoding Time [s]	4	0.2023	0.2110	0.2272	0.1950	Decoding Time [s]	4	0.2106	0.2018	0.2078	0.1997
	5	0.2278	0.2029	0.2078	0.2096		5	0.2291	0.2076	0.1970	0.2152
	6	0.2624	0.2173	0.2230	0.2273		6	0.2057	0.2124	0.1964	0.2003
	7	0.2388	0.2115	0.2162	0.1970		7	0.2082	0.2053	0.1983	0.2041
Volume [byte]	4	1420	1153	915	704	Volume [byte]	4	1402	1143	909	702
	5	1206	939	701	490		5	1188	929	695	488
	6	1200	934	696	485		6	1184	926	692	485
	7	1204	937	699	488		7	1183	924	690	484

Pepper	N	N-K			
		0	1	2	3
PSNR [dB]	4	32.6169	30.3128	30.0602	29.1197
	5	32.0963	29.9893	29.7612	28.8955
	6	32.0913	29.9865	29.7585	28.8929
	7	32.0914	29.9867	29.7588	28.8932
Encoding Time [s]	4	14.7880	2.3244	0.6564	0.2304
	5	14.3656	2.3847	0.7095	0.2876
	6	14.4174	2.4086	0.7376	0.3121
	7	14.5826	2.4150	0.7427	0.3207
Decoding Time [sec]	4	0.2103	0.2026	0.1971	0.2081
	5	0.2246	0.2147	0.1985	0.2220
	6	0.2037	0.1981	0.1977	0.1986
	7	0.2061	0.2201	0.2230	0.1985
Volume [byte]	4	1412	1143	911	702
	5	1198	929	697	488
	6	1206	937	705	496
	7	1209	940	708	499

Let $I_0 = [Y_0\ Cb_0\ Cr_0] \in R^{m \times 3n}$ and $I_k = [Y_k\ Cb_k\ Cr_k] \in R^{m_k \times 3n_k}$ $(k = 1, 2, \cdots, N)$ denote the original image and kth reduced image, respectively, and $N \geq 1$ is a given constant. Y_k is obtained by simply multiple downsampling as follows,

$$(Y_k)_{i,j} = (Y_{k-1})_{2i,2j},\quad i = 1, \cdots, \lfloor \frac{m_{k-1}}{2} \rfloor, j = 1, \cdots, \lfloor \frac{n_{k-1}}{2} \rfloor, \tag{11}$$

where m_k and n_k denote horizontal and vertical sizes of the kth reduced image. Next, we consider a way to generate Cb_k and Cr_k. Since Cb_k and Cr_k are composed by the same method, this paper gives a way to make the reduced image Cb_k. The chrominance images of the kth reduced image are generated so that the $(k-1)$th chrominance image Cb_{k-1} can be restored preciously using Y_{k-1} and Cb_k. Let $\acute{\mathbf{c}}^*_{k-1} \in R^{m_k n_k}$ and $\mathbf{c}^*_{k-1} \in R^{m_{k-1}n_{k-1}-m_k n_k}$ denote vectors formed by pixels of even numbered columns and rows, that is, they do not include pixels of odd numbered columns nor rows. Then, this paper proposes the chrominance image recovery problem of the kth reduced image as follows,

$$\underset{\boldsymbol{x}_k}{\text{Minimize}}\ \|\boldsymbol{c}^*_{k-1} + \left(H^{*T}H^*\right)^{-1} H^{*T}\acute{H}^*\boldsymbol{x}_k\|_2^2 + \lambda\|\acute{\boldsymbol{c}}^*_{k-1} - \boldsymbol{x}_k\|_2^2, \tag{12}$$

Algorithm 1 RP extraction algorithm.

Require: $Y_0, Cb_0, Cr_0, N, K, \varepsilon, t_{max}$
  Generate Y_k $k = 1, 2, \ldots, N$ according to (11).
  Calculate Cb_k, Cr_k for $k = 1, 2, \ldots, N$ according to (14).
  Set $\Omega_N^{RP} \leftarrow \{(i, j, c_b, c_r)|\ i \in \{1, 2, \ldots, m_N\},\ j \in \{1, 2, \ldots, n_N\}, c_b = (Cb_N)_{i,j}, c_r = (Cr_N)_{i,j}\}$.
  for k = N-1,N-2,..., N-K **do**
    $\Omega_k^{ini} \leftarrow \{(i, j)|\ i \in \{2, 4, \ldots, 2m_{k+1}\},\ j \in \{2, 4, \ldots, 2n_{k+1}\}\}$.
    $\Omega_k \leftarrow \Omega_k^{ini}$.
    $t \leftarrow 1$.
    repeat
      Calculate $\bar{Cb}_k$ and $\bar{Cr}_k$ from Y_k and (i, j)-elements of Cb_k and Cr_k for all $(i, j) \in \Omega_k$ using (7).
      for all (i,j) **do**
        $(E)_{i,j} \leftarrow |(Cb_k - \bar{Cb}_k)_{i,j}| + |(Cr_k - \bar{Cr}_k)_{i,j}|$.
      end for
      $(i^{RP}, j^{RP}) \leftarrow \arg\max_{(i,j)}(E_{i,j})$.
      $\Omega_k \leftarrow \Omega_k \cup \left\{\left(i^{RP}, j^{RP}\right)\right\}$.
      $t \leftarrow t + 1$.
    until $t = t_{max}$ or $\max(E_{i,j}) \leq \varepsilon$
    $\Omega_k \leftarrow \Omega_k \setminus \Omega_k^{ini}$.
    $\Omega_k^{RP} = \{(i, j, c_b, c_r)|(i, j) \in \Omega_k, c_b = (Cb_k)_{i,j}, c_r = (Cr_k)_{i,j}\}$.
  end for
Ensure: Ω_k^{RP} $k = N, N-1, \ldots, N-K$

where $\lambda > 0$ is a given constant. Because the chrominance images Cb_{k-1} and Cr_{k-1} are recovered using (7) with Y_{k-1}, Cb_k and Cr_k in the colorization phase as written Section 2.4, this problem gives Cb_k to achieve a good colorization of the $(k-1)$th image. However, problem (12) requires high computational cost to obtain the optimal solution since it takes high cost to calculate $\left(H^{*T}H^*\right)^{-1}$. Therefore this paper provides its relaxed problem to obtain an approximate solution with low computational cost. Let J denote the first term of the objective function in (12). Then, we have that

$$\begin{aligned} J &= \|\boldsymbol{c}_{k-1}^* + \left(H^{*T}H^*\right)^{-1} H^{*T}\acute{H}^*\boldsymbol{x}_k\|_2^2 \\ &= \|\left(H^{*T}H^*\right)^{-1} H^{*T}\left(H^*\boldsymbol{c}_{k-1}^* + \acute{H}^*\boldsymbol{x}_k\right)\|_2^2 \\ &\leq \|\left(H^{*T}H^*\right)^{-1} H^{*T}\|_F^2\ \|H^*\boldsymbol{c}_{k-1}^* + \acute{H}^*\boldsymbol{x}_k\|_2^2. \end{aligned} \tag{13}$$

Since J is bounded by $\|H^*\boldsymbol{c}_{k-1}^* + \acute{H}^*\boldsymbol{x}_k\|_2^2$, we consider the following problem to reduce J instead of (12),

$$\underset{\boldsymbol{x}_k}{\text{Minimize}} \quad \|H^*\boldsymbol{c}_{k-1}^* + \acute{H}^*\boldsymbol{x}_k\|_2^2 + \lambda\|\acute{\boldsymbol{c}}_{k-1}^* - \boldsymbol{x}_k\|_2^2. \tag{14}$$

Utilizing (11) and (14) to obtain a set of multiple resolution images, this paper provides the RPs extraction method as shown in Algorithm 1, where $\bar{Cb}_k$ and $\bar{Cr}_k$ denote the colorized chrominance images of I_k. This algorithm extracts RPs from I_N to I_{N-K} in turn for given constant $K \geq 1$, and the even numbered pixels of I_k are used as initial RPs at each iteration. The performance of Algorithm 1 depends heavily on the value of K. If the value of K is large, lots of RPs are extracted, that is, the K determines the number of RPs, the calculation time and the volume of information to store RPs. The numerical examples in Section 3 show the effects of K.

Note that the chrominance values of these even numbered pixels are deleted in Ω_k^{RP} at the end of each iteration. They are restored as Cb_{k+1} and Cr_{k+1} in the previous iteration and are reused as RPs for recovering I_k. Therefore Algorithm 1 does not store them in Ω_k^{RP} to reduce the amount of information about RPs. All pixels of the Nth reduced image are members of Ω_N^{RP}. Figure 3 shows outline of the algorithm.

Algorithm 2 Colorization algorithm.

Require: Y_0, Ω_k^{RP} $(k = N, N-1, \ldots, N-K)$
  Generate Y_k for $k = 1, 2, \ldots, N$ by (11).
  $\Omega_k^{RP} \leftarrow \{\phi\}$ for $k = N-K-1, \ldots, 0$.
  Generate Cb_N and Cr_N from Ω_N^{RP}.
  for k= $N-1$,N-2,...,0 **do**
    $\Omega_k^{RP_{even}} = \{(2i, 2j, c_b, c_r)|i \in \{1, 2, \ldots, m_{k+1}\},\ j \in \{1, 2, \ldots, n_{k+1}\}, c_b = (Cb_{k+1})_{i,j}, c_r = (Cr_{k+1})_{i,j}\}$.
    $\Omega_k^{RP} \leftarrow \Omega_k^{RP} \cup \Omega_k^{RP_{even}}$.
    Recover the chrominance images Cb_k and Cr_k with Ω_k^{RP} by (7).
  end for
Ensure: colorized image.

2.4 Color image recovery

This subsection provides the chrominance image recovery algorithm. Now we consider a way to recover an image $I_0 = [Y_0\ Cb_0\ Cr_0] \in R^{m \times 3n}$ from the stored RPs Ω_k^{RP} $(k = N, \ldots N-K)$. First, I_N is completely obtained by Ω_N^{RP}. Next, we focus onto recover the image I_k $(k = N-1, N-2, \ldots, 0)$. Since Ω_k^{RP} includes RPs without the even numbered pixels, image I_{k+1} is used as Ω_k^{RP} corresponding to the even numbered pixels. Then the color images I_k is restored by (7) using Ω_k^{RP} and I_{k+1}. Finally we obtain the colorization algorithm with multiple resolution images as shown in Algorithm 2, and Fig. 4 shows its outline.

The calculation cost of the colorization proposed in subsection 2.2 mostly depends on the calculating of inverse matrix $\left(H^{*T}H^*\right)^{-1}$, and the size of $\left(H^{*T}H^*\right)^{-1}$ depends on the number of unknown color pixels. The number of unknown color pixels in Algorithm 2 is equal to the following,

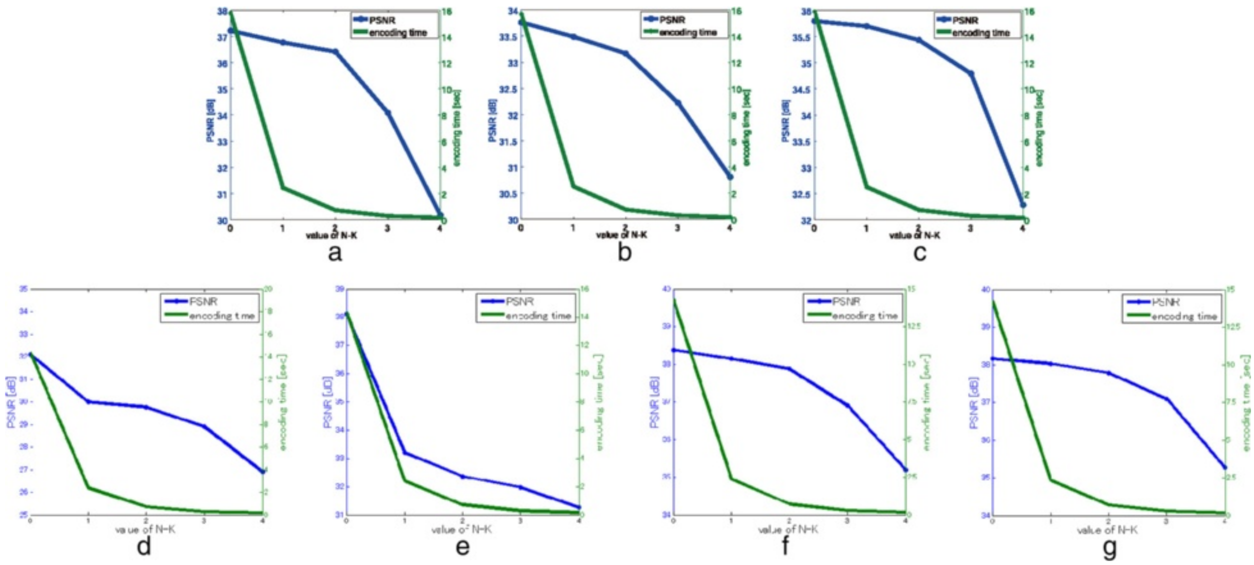

Fig. 6 The value of $N - K$ vs. PSNR and coding time for **a** milkdrop, **b** boat, **c** lenna, **d** pepper, **e** airplane, **f** couple, and **g** earth

$$\sum_{k=0}^{N-1} \left(m_k \times n_k - m_{k+1} \times n_{k+1} - |\Omega_k^{RP}|\right)$$
$$= m_0 \times n_0 - m_N \times n_N - \sum_{k=0}^{N-1} \left|\Omega_k^{RP}\right| \tag{15}$$
$$= m_0 \times n_0 - \sum_{k=0}^{N} \left|\Omega_k^{RP}\right|.$$

This implies that the calculation cost does not depend on N. Numerical examples in Section 3 show that the computing time is independent of N.

2.5 Volume of information of RPs

This subsection gives an encoding method of RPs.

Let us consider the case that an $m \times n$ 24-bit image is compressed with p RPs. If the information of RPs is composed of their coodinates and two chrominance values,

Fig. 7 Partial test images (192 × 192): **a** milkdrop, **b** boat, **c** lenna, **d** pepper, **e** airplane, **f** couple, and **g** earth

$p(2 \times 8 + \lceil \log_2 mn \rceil)$ [bits] are required to represent all RPs because 2×8 [bits] and $\lceil \log_2 mn \rceil$ [bits] are required to represent their chrominance values and coordinates, respectively. The proposed algorithm extracts RPs based on multiple resolution images and requires less information to represent them.

We consider encoding of the coordinates of RPs Ω_k^{RP} $(k = N, N-1, \ldots N-K)$. In Algorithm 1, Ω_N^{RP} is given as the reduced image I_N, the Ω_N^{RP} are given as the reduced image I_N, that is, all pixels of I_N are RPs. Therefore, the coordinates of Ω_N^{RP} are not required, and only the value of N is coded to represent Ω_N^{RP}. Next, we consider coding of the coordinates of RPs in Ω_k^{RP} $(k = N-1, \ldots, N-K)$. For $m \times n$ image, we store the coordinates of RPs as a rasterized vector of a binary matrix of size $m \times n$ whose entries are 1 if corresponding pixels are RPs and 0 if otherwise. Hence, this vector is sparse, and it is coded by run length encoding (RLE), which stores the length of consecutive 0s. Empirical results show that the length of consecutive 0s on the image Ω_{N-k}^{RP} is less than z_k defined by

$$z_{N-k} = \frac{m_{N-k}n_{N-k} - m_{N-k+1}n_{N-k+1}}{t_{max}}. \tag{16}$$

Therefore, this paper proposes to encode the coordinate by RLE per following bit,

$$bitunit_{N-k} = \max\left(\left\lceil \log_2(z_{N-k}) \right\rceil, 2\right). \tag{17}$$

Table 1 shows the coding table of $bitunit_{N-k} = 3$. This decoding method requires the run length data of Ω_k^{RP} $(k = N-1, \ldots, N-K)$ and the values of N and t_{max} to decode the coordinates of RPs.

3 Numerical examples

This section presents numerical examples to show the efficiency of the proposed algorithm. We use the test

Table 3 Numerical results of comparison with other colorization based image coding algorithms

Image	Algorithm	Number of RPs [pixels]	Volume of information [byte]	Time [sec]		PSNR [dB]		SSIM	
				encoding	decoding	Cb	Cr	Cb	Cr
Milkdrop	algorithm proposed in [9]	109	436	1.31×10^3	3.51	35.0257	27.1737	0.9232	0.8424
	random algorithm	108	432	—	0.17	33.2654	27.5243	0.9298	0.8818
	proposed algorithm with (12)	108	273	4.17×10^3	0.11	38.2353	33.7145	0.9574	0.9332
	proposed algorithm with (14)	108	273	0.23	0.11	37.9307	34.5197	0.9588	0.9371
Boat	algorithm proposed in [9]	117	468	1.19×10^3	3.53	33.9007	30.2299	0.8798	0.8216
	random algorithm	117	468	—	0.17	32.5359	28.6484	0.8766	0.8173
	proposed algorithm with (12)	117	308	4.02×10^3	0.12	34.4766	32.3504	0.8962	0.8774
	proposed algorithm with (14)	117	308	0.25	0.12	33.9637	32.1755	0.8946	0.8802
Lenna	algorithm proposed in [9]	117	468	1.20×10^3	3.57	33.6590	32.3297	0.8665	0.8335
	random algorithm	117	468	—	0.18	31.1721	30.8207	0.8546	0.8344
	proposed algorithm with (12)	117	310	4.02×10^3	0.12	34.3554	33.1450	0.8860	0.8660
	proposed algorithm with (14)	117	310	0.24	0.12	33.9673	32.8775	0.8846	0.8662
Pepper	algorithm proposed in [9]	113	452	1.25×10^3	3.50	31.8969	26.8608	0.8555	0.8260
	random algorithm	111	444	—	0.1716	29.9528	24.5312	0.8536	0.8209
	proposed algorithm with (12)	111	285	3.90×10^3	0.12	33.2907	29.4019	0.8950	0.8809
	proposed algorithm with (14)	111	284	0.24	0.12	32.9208	30.2535	0.8930	0.8852
Airplane	algorithm proposed in [9]	119	476	1.21×10^3	3.55	31.0753	34.4833	0.8166	0.9014
	random algorithm	117	468	—	0.17	29.8990	32.9087	0.8169	0.9035
	proposed algorithm with (12)	117	307	3.90×10^3	0.12	33.7608	36.1246	0.8879	0.9154
	proposed algorithm with (14)	117	307	0.27	0.12	33.8210	35.7528	0.8883	0.9155
Couple	algorithm proposed in [9]	118	472	1.26×10^3	3.47	35.9439	33.0840	0.9001	0.8795
	random algorithm	117	468	—	0.17	34.2172	31.2331	0.8992	0.8804
	proposed algorithm with (12)	117	311	3.90×10^3	0.14	37.4989	35.2208	0.9210	0.9120
	proposed algorithm with (14)	117	309	0.25	0.13	37.1587	35.2879	0.9188	0.9115
Earth	algorithm proposed in [9]	115	460	1.19×10^3	3.56	36.1886	33.8077	0.9133	0.8290
	random algorithm	114	456	—	0.17	34.1523	32.4901	0.9068	0.8277
	proposed algorithm with (12)	114	288	3.90×10^3	0.12	37.6137	34.4186	0.9386	0.8598
	proposed algorithm with (14)	114	289	0.26	0.12	37.9318	34.1140	0.9390	0.8602

images as shown in Fig. 5. In order to evaluate the quality of image compression, we measure the differences between the original image and the recovered image using the peak signal to noise ratio (PNSR) and the structural similarity (SSIM) [16]. In order to calculate these evaluates, we use the MATLAB functions `psnr` and `ssim`. In all experiments we use $\varepsilon = 2$ and $\lambda = 0.3$, which are selected empirically from $\varepsilon \in [0, 10]$ and $\lambda \in [0.1, 1]$ to achieve the best performance. We use $t_{max} = 70$ except for the second experiment.

First we examine the effect of the parameters N and K using uncompressed luminance images. Table 2 shows PSNR, the computing time of Algorithm 1, the computing time of Algorithm 2 and the volume of information to store RPs. PSNR is obtained by averaging those of Cb and Cr. As can be seen, the results of $N \geq 5$ are almost the same. Figure 6 shows the PSNR and the computing time of Algorithm 1 with $N = 5$ for $N - K \in \{0, 1, 2, 3, 4\}$. We can see that the algorithm with $(N, K) = (5, 3)$ achieves the best tradeoff between calculation time and PSNR, and therefore we use $(N, K) = (5, 3)$ in the rest of this section.

Next this paper shows the coding performance of the proposed algorithm comparing with the algorithm proposed in [9] and the random algorithm which selects RPs randomly. We chose $\lambda = 5$ in (12), which is selected empirically from $\lambda \in [1, 10]$ to achieve the best performance. To evaluate the performance of the relaxed problem (14) in RPs extraction scheme, we also examine Algorithm 1 with (12) instead of (14). Because we cannot fix the number of RPs exactly in the algorithm proposed in [9], we adjust the parameter t_{max} of Algorithm 1 such that the number of RPs is nearly equal to that of [9]. The random algorithm selects RPs randomly and gives the same number of RPs as the proposed algorithm, and we use colorization algorithm (7). In order to see the performance of colorization based coding method, these algorithms use uncompressed luminance images. Because the algorithm proposed in [9] and the proposed algorithm with (12) use a large amount of memory and take a high computational cost, they cannot be applied to a whole image of the test images. Therefore, we use their partial images as shown in Fig. 7. Table 3 shows the number of RPs, the volume of information to restore the RPs, SSIM, PSNR, the computing time to extract the RPs (encoding

Table 4 Numerical results of comparison with Lee's algorithm

Image	Algorithm	Number of RPs [pixels]	Volume of information [byte]	Time [sec]		PSNR [dB]		SSIM	
				encoding	decoding	Cb	Cr	Cb	Cr
Milkdrop	proposed algorithm	274	699	0.72	0.22	37.8741	35.8627	0.9635	0.9377
	algorithm proposed in [12]	274	959	81.78	31.01	37.1474	32.4973	0.9432	0.9049
	algorithm proposed in [12]	200	700	53.41	29.27	36.8288	32.4900	0.9397	0.8978
Boat	proposed algorithm	274	698	0.71	0.22	34.7794	32.0441	0.8928	0.8734
	algorithm proposed in [12]	274	959	79.87	28.77	34.5627	32.8473	0.8815	0.8744
	algorithm proposed in [12]	200	700	52.57	28.66	34.1386	32.5993	0.8753	0.8707
Lenna	proposed algorithm	274	701	0.71	0.21	36.1906	35.3057	0.9231	0.9018
	algorithm proposed in [12]	274	959	80.88	29.55	36.2129	34.5499	0.9086	0.8925
	algorithm proposed in [12]	200	700	53.39	29.51	35.9824	34.5354	0.9059	0.8903
Pepper	proposed algorithm	274	697	0.71	0.20	29.4435	30.0789	0.8697	0.8762
	algorithm proposed in [12]	274	959	88.56	38.37	32.6670	29.7992	0.8702	0.8510
	algorithm proposed in [12]	199	697	59.50	37.39	30.2358	28.8093	0.8458	0.8256
Airplane	proposed algorithm	274	702	0.71	0.22	33.0319	31.6955	0.9117	0.9149
	algorithm proposed in [12]	274	959	79.14	29.15	34.9049	33.9563	0.9124	0.9185
	algorithm proposed in [12]	200	700	52.83	29.07	34.7446	33.8131	0.9100	0.9165
Couple	proposed algorithm	274	703	0.70	0.20	38.7016	37.0570	0.9216	0.9172
	algorithm proposed in [12]	274	959	84.51	34.82	38.4629	35.9642	0.9143	0.9013
	algorithm proposed in [12]	201	704	56.13	32.76	38.3394	35.8256	0.9134	0.9012
Earth	proposed algorithm	274	695	0.71	0.21	39.4627	36.1058	0.9500	0.9049
	algorithm proposed in [12]	274	959	85.51	35.81	40.2759	37.7125	0.9536	0.9166
	algorithm proposed in [12]	199	697	58.08	35.84	40.0729	37.6271	0.9525	0.9156

time), and to colorize the images (decoding time). As can be seen, the proposed algorithm with (14) works faster and achieves better performance than the other algorithms.

Thirdly, the proposed algorithm is compared with the algorithm proposed in [12] using the test images as shown in Fig. 5. In [12], Lee et al. have proposed the colorization-based image coding algorithm, which achieves a high coding performance. In order to see the performance of colorization-based coding method, these algorithms use uncompressed luminance images. Since an arbitrary number of RPs can be used in Lee's algorithm, we examine the algorithm in two ways by using (1) the same number of RPs as those of the proposed algorithm and (2) RPs whose volume of information to be represented is equal to that of the proposed algorithm.

Table 4 shows the number of RPs, the volume of information to restore the RPs, SSIM, PSNR, the computing time to extract the RPs (encoding time) and to colorize the images (decoding time). We can see that the proposed algorithm achieves similar performance to Lee's algorithm and requires less computational.

Finally, we compare the proposed algorithm with JPEG and JPEG2000 coding method. Luminance images are compressed using JPEG/JPEG2000 coding where the quality parameter (QP) is selected such that the volume of information is equal to about 4000 bytes.

Table 5 shows the volume of information [byte] to represent each color image, SSIM, PSNR and quality parameter (QP) for JPEG/JPEG2000, Figs. 8, 9 and 10 show the recovery images, and Fig. 11 shows zoomed images of Fig. 8. SSIM and PSNR are obtained by averaging those of red, green, and blue images. As can be seen, the proposed algorithm with JPEG2000 can compress color images the best of all while the qualities of compressed images are equal or better than other coding methods. The proposed

Table 5 Numerical results of comparison with JPEG and JPEG2000

Image	Algorithm	Volume of information [byte]	PSNR[dB]	SSIM	QP for JPEG/JPEG2000
Milkdrop	JPEG	4206	28.6925	0.7988	20
	JPEG2000	4094	32.4263	0.8719	46
	proposed algorithm with JPEG	4138	30.7470	0.8540	22
	proposed algorithm with JPEG2000	4036	32.6914	0.8891	20
Boat	JPEG	4139	24.7191	0.7488	10
	JPEG2000	4068	26.9982	0.8219	46
	proposed algorithm with JPEG	4125	25.8622	0.7849	10
	proposed algorithm with JPEG2000	4039	27.2043	0.8374	20
Lenna	JPEG	4136	26.794	0.7864	13
	JPEG2000	4070	29.2353	0.8398	46
	proposed algorithm with JPEG	4159	27.6452	0.8116	13
	proposed algorithm with JPEG2000	4061	29.2051	0.8568	20
Pepper	JPEG	4137	24.0496	0.7078	11
	JPEG2000	4071	27.1189	0.7969	46
	proposed algorithm with JPEG	4147	24.8098	0.7632	12
	proposed algorithm with JPEG2000	4051	25.9898	0.8127	20
Airplane	JPEG	4176	25.1615	0.7894	12
	JPEG2000	4062	28.0905	0.8443	46
	proposed algorithm with JPEG	4192	25.6759	0.8108	12
	proposed algorithm with JPEG2000	4044	26.7533	0.8471	20
Couple	JPEG	4158	30.5207	0.8112	21
	JPEG2000	4018	32.0973	0.8373	46
	proposed algorithm with JPEG	4184	31.0459	0.8314	21
	proposed algorithm with JPEG2000	4064	32.1436	0.8527	20
Earth	JPEG	4277	27.5088	0.7952	13
	JPEG2000	4164	30.0662	0.8527	45
	proposed algorithm with JPEG	4146	28.4777	0.8088	12
	proposed algorithm with JPEG2000	4017	30.2628	0.8604	20

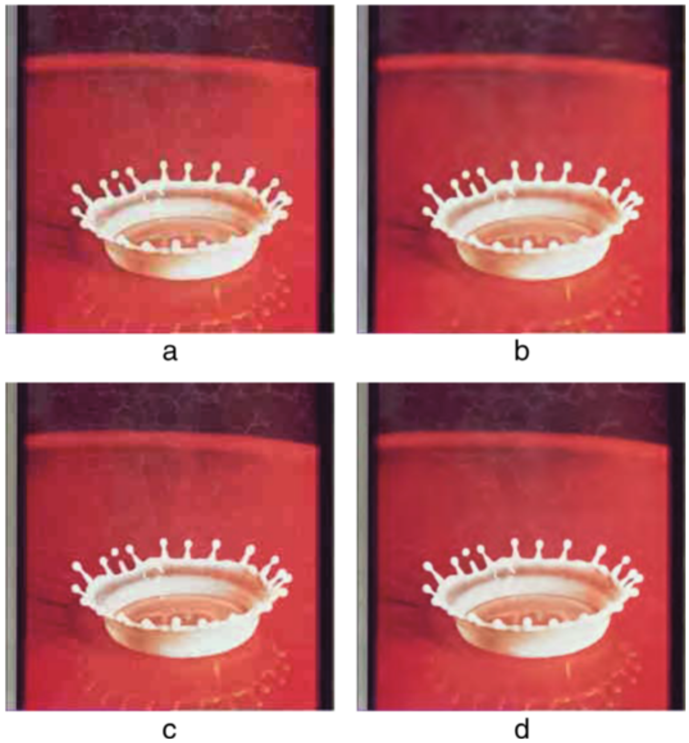

Fig. 8 Image results (milkdrop) of **a** JPEG, **b** JPEG2000, **c** proposed algorithm with JPEG, and **d** proposed algorithm with JPEG2000

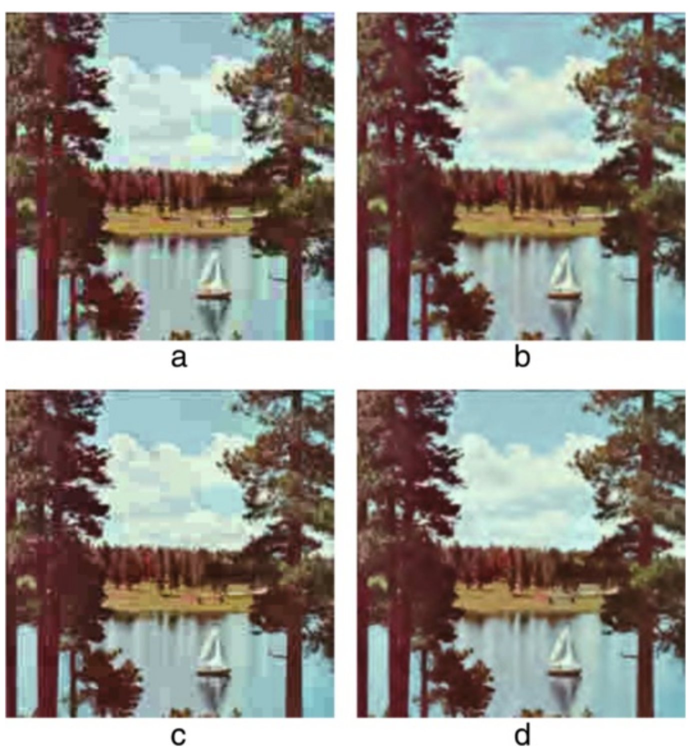

Fig. 9 Image results (boat) of **a** JPEG, **b** JPEG2000, **c** proposed algorithm with JPEG, and **d** proposed algorithm with JPEG2000

Fig. 10 Image results (lenna) of **a** JPEG, **b** JPEG2000, **c** proposed algorithm with JPEG, and **d** proposed algorithm with JPEG2000

algorithm requires 0.7–0.8 [s] for encoding and 0.2–0.3 [s] for decoding in MATLAB 2014a on a PC with an Intel Core i7 3.4 GHz CPU, 8 GB of RAM memory.

4 Conclusions

This paper proposes a representative pixel (RP) extraction and colorization algorithm for the colorization-based digital image coding. In order to achieve low computing cost and high image coding performance, multiple reduced images are generated by colorization error minimizing method, and the RPs are extracted from these reduced images. Numerical examples show that the proposed algorithm can extract RPs and recover a color image fast and effectively

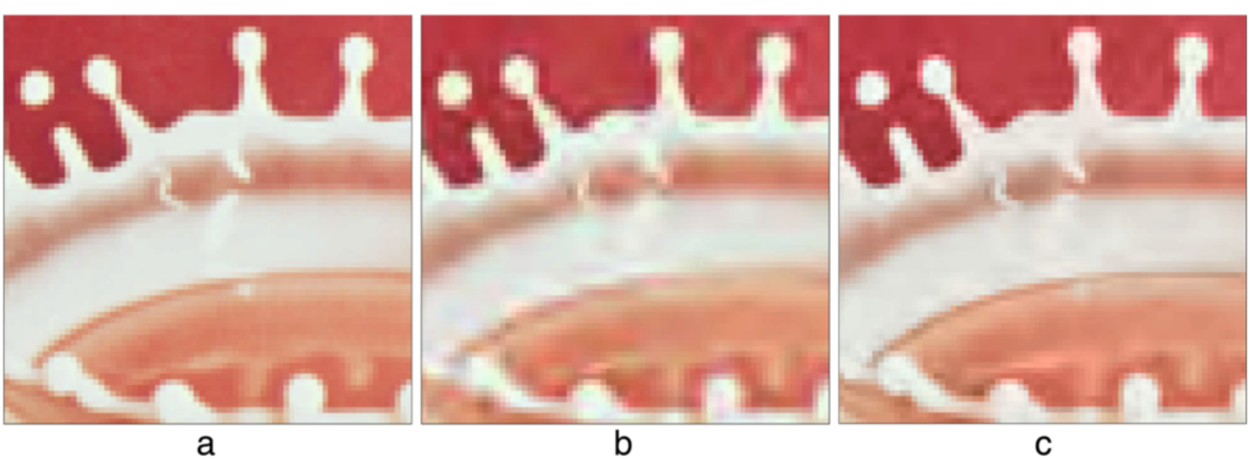

Fig. 11 Zoomed image results (milkdrop) of **a** original, **b** JPEG2000, and **c** proposed algorithm with JPEG2000

comparing with other colorization-based algorithms, and numerical results show that the proposed algorithm achieves higher coding performance than JPEG/JPEG2000.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
We are grateful to Sukho Lee and his co-authors for providing the source code of the colorization-based coding proposed in [12].
This work was supported by JSPS KAKENHI Grant Numbers 26·6546.

Author details
[1]Tokyo University of Science, 162-8677 Tokyo, Japan. [2]Kogakuin University, Tokyo, Japan.

Received: 9 January 2015 Accepted: 1 February 2016
Published online: 10 February 2016

References
1. A Levin, D Lischinski, Y Weiss, Colorization using optimization. ACM Trans. Graph. **23**(3), 689–694 (2004)
2. L Yatziv, G Sapiro, Fast image and video colorization using chrominance blending. IEEE Trans. Image Process. **15**(5), 1120–1129 (2006)
3. J Pang, OC Au, K Tang, Y Guo, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Image colorization using sparse representation, (2013), pp. 1578–1582
4. K Uruma, K Konishi, T Takahashi, T Furukawa, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Image colorization algorithm using series approximated sparse function, (2013), pp. 1215–1219
5. T Miyata, Y Komiyama, Y Inazumi, Y Sakai, in *Proc. Picture Coding Symp.* Novel inverse colorization for image compression, (2009), pp. 1–4
6. H Noda, N Takao, M Niimi, in *Proc. IEEE Int. Conf. Image Process. (ICIP)*. Colorization in YCbCr space and its application to improve quality of JPEG Color Images, (2007), pp. 385–388
7. L Cheng, S Vishwanathan, in *Proc. Int. Conf. Mach. Learn. (ICML)*. Learning to compress images and videos, (2007), pp. 161–168
8. X He, M Ji, H Bao, in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*. A unified active and semi-supervised learning framework for image Compression, (2009), pp. 65–72
9. S Ono, T Miyata, Y Sakai, in *Proc. Picture Coding Symp*. Colorization-based coding by focusing on characteristics of colorization bases, (2010), pp. 230–233
10. Y Inoue, T Miyata, Y Sakai, Colorization based image coding by using local correlation between luminance and chrominance. IEICE Trans. Inf. Syst. **95**(1), 247–255 (2012)
11. T Ueno, T Yoshida, M Ikehara, *Color image coding based on the colorization,* Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), (2012), pp. 1–4
12. S Lee, S Park, P Oh, M Kang, Colorization-based compression using optimization. IEEE Trans. Image Proc. **22**(7), 2627–2636 (2013)
13. EJ Candes, J Romberg, T Tao, Stable signal recovery for incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(3), 1207–1223 (2013)
14. DL Donoho, Compressed sensing. IEEE Trans. Inf. Theory. **52**, 1289–1306 (2006)
15. C Rusu, SA Tsaftaris, in *Proc. Int. Symp. Image and Signal Processing and Analysis (ISPA)*. Estimation of scribble placement for painting colorization, (2013), pp. 564–569
16. Z Wang, AC Bovik, HR Sheikh, EP Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Proc. **13**(4), 600–612 (2004)