

# Application of Monte Carlo simulation with block-spin transformation based on the Mumford–Shah segmentation model to three-dimensional biomedical images



Satoshi Sashida <sup>a,\*</sup>, Yutaka Okabe <sup>a</sup>, Hwee Kuan Lee <sup>b</sup>

<sup>a</sup> Department of Physics, Tokyo Metropolitan University, Hachioji, Tokyo 192-0397, Japan

<sup>b</sup> Bioinformatics Institute, 30 Biopolis Street, #07-01 Matrix, 138671, Singapore

## ARTICLE INFO

### Article history:

Received 12 September 2015

Revised 5 June 2016

Accepted 8 June 2016

Available online 9 June 2016

### Keywords:

Mumford–Shah segmentation model

Image segmentation

Monte Carlo method

Graph cuts

Three-dimensional biomedical images

## ABSTRACT

In this paper, we present the iterative Monte Carlo method for solving Mumford–Shah segmentation model in the case of three-dimensional images with emphasis on multi-phase segmentation. The present method introduces iterative descent process to the preceding Monte Carlo method, proposed by our group, to improve convergence. The numerical simulations have shown that the present method overcomes the problem of the preceding Monte Carlo method that converges to local minima in some cases. The computational time of the present method can be shortened by introducing block-spin transformation procedure. We have also compared the result of the present method with the graph cuts method. The comparison has shown that the proposed method converges to almost the same solution of the graph cuts method in reasonably short time, and is superior in memory consumption, especially in the case of multi-phase segmentation. The comparison of the output pattern with the clinical experts' annotation suggests that the Mumford–Shah segmentation model is suitable for a multi-phase image segmentation model of biomedical images. Because of the advantage of small memory consumption, the present Monte Carlo method with the block-spin transformation can be applied to a wide range of three-dimensional images. We make a remark that the block-spin transformation is also applicable to the graph cuts method, which leads to the saving of the computational time while maintaining lower-energy convergence.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction

Among various active topics in image processing such as noise removal, object recognition, image registration etc., we focus on image segmentation problem in this paper. Image segmentation is the process that divides an image into multiple parts, sets of pixels. A typical purpose of image segmentation is pre-processing that simplifies an image into a few number of meaningful parts that can be easily analyzed in subsequent processes, such as object recognition, video tracking, image matching and surface reconstruction. Since the performance of the image segmentation affects the quality of subsequent quantitative descriptions, or visualizations, image segmentation is very important in many cases.

Especially in biomedical fields, three-dimensional image processing has captured a lot of interest. Because of the rapid progress of imaging apparatuses, we should treat hundreds of CT (Computed Tomography) or MR (Magnetic Resonance) images of larger

sizes in recent years. Faster fully-automated or semi-automated image processing is needed, especially for three-dimensional images.

The Mumford–Shah segmentation model ([Mumford and Shah, 1989](#)) is an energy model for image segmentation in the field of computer vision. By exploring lower-energy solution, an output image with piecewise constant regions and smooth contour is obtained. Mumford–Shah segmentation model is suitable for an image including the object with a blurred outline and an image containing noise. In this study, we focus on the three-dimensional image segmentation problem based on the Mumford–Shah segmentation model.

There have been many works for solving the two-dimensional Mumford–Shah segmentation model. The level set method approach by [Chan and Vese \(2001\)](#) had great success for binary segmentation. The level set method ([Osher and Fedkiw, 2003](#)) is a kind of dynamic contour model where a contour is represented implicitly as the zero level set of a scalar Lipschitz continuous function  $\phi$ , which is called level set function. In the level set method, a contour of the Mumford–Shah segmentation model is formulated by level set function. An output image is obtained by solving as-

\* Corresponding author.

E-mail addresses: [sashida\\_s@phys.se.tmu.ac.jp](mailto:sashida_s@phys.se.tmu.ac.jp) (S. Sashida), [okabe@phys.se.tmu.ac.jp](mailto:okabe@phys.se.tmu.ac.jp) (Y. Okabe), [leehk@bii.a-star.edu.sg](mailto:leehk@bii.a-star.edu.sg) (H.K. Lee).

sociated gradient descent equations numerically. The extension of level set method to multi-phase segmentation was proposed (Vese and Chan, 2002). A variant of level set method, the piecewise constant level set method (PCLSM), was also proposed (Lie et al., 2005; 2006a; 2006b). Different from standard level set method, any partition can be described by single level set function which takes discrete values on each region, and the length of contour is approximated by the total variation of the level set function.

The solution of the Mumford–Shah segmentation model using graph cuts method (Boykov and Kolmogorov, 2004; Boykov et al., 2001; Greig et al., 1989; Kolmogorov and Zabih, 2004), which is well known in the field of computer vision, has captured a lot of attention. By finding a minimum cut in a graph corresponding to the energy model, the minimum energy solution can be computed. The application to the two-phase Mumford–Shah segmentation model was reported: it was shown that the graph cuts method is much more efficient and robust than the gradient descent minimization (Darbon, 2007; El-Zehiry et al., 2007; Zeng et al., 2006). The extension to multi-phase segmentation was proposed by El-Zehiry and Elmaghraby (2007). In their method, each phase is segmented iteratively by the two-label graph cuts method. Bae and Tai (2008) and Bae and Tai (2009) proposed multi-phase graph cuts method which is based on the work of Ishikawa and Geiger (1998), Ishikawa (2003) and Darbon and Sigelle (2006) under the framework of PCLSM. The level set formulation of the Mumford–Shah segmentation model is related to the multi-label graph, and the solution is obtained by calculating a minimum cut iteratively. The efficiency of the graph cuts method was reported to be hundreds of times faster than the previous gradient-descent based approach (Lie et al., 2006b) and insusceptible to initial configuration while maintaining the lower-energy outputs (Bae and Tai, 2008, 2009). Gurholt and Tai extended the work of PCLSM-based graph cuts approach (Bae and Tai, 2008; 2009) to three-dimensional image segmentation problem (Gurholt and Tai, 2009).

On the other hand, the solution of the Mumford–Shah segmentation model using the Monte Carlo method was proposed (Watanabe et al., 2011). The Monte Carlo method is a probabilistic numerical approach, which is a standard simulation method for many particle systems in statistical physics. By mapping the data of image to the highly correlated lattice model, the Monte Carlo method can be applied in the field of computer vision. Geman and Geman applied Monte Carlo technique in the image processing by simulated annealing (Geman and Geman, 1984). Tu and Zu have applied the data-driven Markov Chain Monte Carlo (DDMCMC) to the color and gray-level image segmentations (Tu and Zhu, 2002). Dellaert et al. proposed the chain flipping Monte Carlo technique combined with EM (Expectation Maximization) algorithm for solving the structure-from-motion (SFM) problem (Dellaert et al., 2003). Zhai and Shah applied the MCMC technique involving jump and diffusion method (Green, 1995) to video scene segmentation (Zhai and Shah, 2006). Erdogan et al. applied generalized Swendse–Wang sampling (Barbu and Zhu, 2005; Swendsen and Wang, 1987) to planar segmentation from RGBD sensors (Erdogan et al., 2012).

In the application to the Mumford–Shah segmentation model, Watanabe et al. (2011) studied the piecewise constant Mumford–Shah energy model by simulated annealing with the Metropolis sampling (Metropolis et al., 1953). With the use of the block-spin transformation, they succeeded in the fast convergence to the low-energy solution. Recently, the hybridization of the graph cuts method and Monte Carlo method for solving multi-phase Mumford–Shah segmentation model was reported by the present authors (Sashida et al., 2014). This hybrid method picks up the advantages of both methods. It is robust to initial conditions, and easily approaches global minimum solution to remove statistical fluctuations. For the three-dimensional images of large size, the hybrid

method is not necessarily effective because of the problem of large memory consumption of the graph cut method, which will be discussed in the present paper.

In this paper, we apply the Monte Carlo method by Watanabe et al. (2011) to solve three-dimensional Mumford–Shah segmentation model. Since the original method sometimes faces the problem of local minimum, we propose a modified Monte Carlo method that includes iterative updating processes to improve the convergence. We do some numerical simulations that observe the convergence of solutions, the values of obtained energy, the computational time, the memory consumption, etc. We show that our proposed Monte Carlo is compatible to the graph cuts method (Gurholt and Tai, 2009) in lower-energy convergence, although small thermal fluctuations remain. We also show that our modified Monte Carlo method can be accelerated by introducing block-spin transformation to iterative procedure with maintaining lower-energy convergence. The computational speed of the block-spin modified Monte Carlo method is comparable to that of the graph cuts method. We also show that our modified Monte Carlo method has advantages in memory consumption that is important in practical use of three-dimensional images; the efficiency of our modified Monte Carlo method is more prominent when the system size of images is larger, and when the number of phases is larger.

This paper is organized as follows. In Section 2, we give an introduction of Mumford–Shah segmentation model used in this paper. In Section 3, we describe two methods of the Monte Carlo-based solutions. One is the method which was proposed previously (Watanabe et al., 2011), and the other is the method which we propose here to improve convergence. In Section 4, we show the numerical results of energy minimization methods applied to three-dimensional brain tumor images. The comparison is made for several methods. Finally, Section 5 is devoted to summary and discussions.

## 2. Energy model

The Mumford–Shah energy model (Mumford and Shah, 1989) is a famous energy model with a wide range of applications in image processing. Let the image domain be  $\Omega$  and intensity of the input image be  $u(x)$ . The image domain  $\Omega$  is partitioned into exclusive segments  $\Omega_l, l = 1, 2, \dots, n$  by  $C$ , which is a set of closed contours of  $\Omega_l$ , so that  $\Omega = \Omega_1 \cup \Omega_2 \cup \dots \cup C$ . Let variables  $\{c\} = c_1, c_2, \dots, c_n$  be constant values on each region  $\Omega_l$ , Mumford–Shah segmentation model can be described as follows,

$$E(\{c\}, C) = \sum_{l=1}^n \int_{\Omega_l} (u(x) - c_l)^2 dx + \mu |C| \quad (1)$$

where  $|C|$  stands for the total length of the contours making up  $C$  and  $\mu$  is smoothing parameter. Once  $E(\{c\}, C)$  is minimized, we obtain the segmented image with  $n$  regions.

In order to solve the Mumford–Shah segmentation model by Monte Carlo method, we rewrite Eq. (1) in three-dimensional discrete lattice. Let  $i = (x, y, z)$  be a grid point on a three-dimensional lattice  $\mathcal{P}$  and define the neighborhood system  $\mathcal{N}(i) = \{(x \pm 1, y, z), (x, y \pm 1, z), (x, y, z \pm 1)\}$ . In order to give  $\Omega_l$  to discrete representation, we introduce variables that represent image segments on discrete lattice as  $\{\phi\}$ , where  $\phi_i = 1, 2, \dots, n$  at the grid point  $i$ . The energy model on a discrete lattice is written as follows:

$$E(\{c\}, \{\phi\}) = \sum_{i \in \mathcal{P}} (u_i - c_{\phi_i})^2 + \mu |C|, \quad (2)$$

$u_i$  is the intensity of the input image at the grid point  $i$ , and  $\{c\}$  are constant values on each region  $\phi_i$ . Now we refer to the first term of Eq. (2) as “the data term” and the second term as “the regularization term”.

There are some choices on the form of the regularization term of Eq. (2): Watanabe et al. (2011) used the isotropic Potts interaction,

$$|C| = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}(i)} \frac{1}{2} \delta_{\phi_i, \phi_j}, \quad (3)$$

while the discrete version of total variation of  $L_1$  norm ( $TV_1$ ) was used in ref. Sashida et al. (2014),

$$|C| = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}(i)} \frac{1}{2} |\phi_i - \phi_j|. \quad (4)$$

In order to compare our proposed method with the three-dimensional graph cuts approach by Gurholt and Tai (2009) in Section 4, we have chosen Eq. (4) as regularization term in this paper. Because of limitation of algorithm in Ishikawa (2003), the graph cuts approach of Gurholt and Tai (2009) can only be applicable to the MRF model with a regularization term that is convex in terms of a linearly ordered label set. So the energy model is finally described as follows:

$$E(\{c\}, \{\phi\}) = \sum_{i \in \mathcal{P}} (u_i - c_{\phi_i})^2 + \mu \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}(i)} \frac{1}{2} |\phi_i - \phi_j|. \quad (5)$$

### 3. Monte Carlo method for Mumford–Shah segmentation model

In this section, we introduce two Monte Carlo-based solutions of the three-dimensional Mumford–Shah segmentation model, Eq. (5). We first explain the three-dimensional extension of the method of Watanabe et al. (2011). Next, we propose an iterative Monte Carlo method for improving convergence.

Watanabe et al. (2011) succeeded in finding a low-energy solution effectively in the case of two-dimensional Mumford–Shah segmentation model. The method of Watanabe et al. is a combination of simulated annealing and the block-spin transformation. It starts from high-temperature state with thermal fluctuations and gradually removing thermal fluctuations; eventually it reaches the global minimum solution escaping from local minima. In actual simulation, because of the finite computation time, it converges to a solution which is very close to the global minimum solution. The transition of states is implemented by the standard Metropolis algorithm (Metropolis et al. 1953). The process of the simulated annealing is described as follows:

#### Algorithm A

**Step 1** Set temperature parameter  $T \leftarrow T^0$ , Monte Carlo Step (or MCS for abbreviation)  $s \leftarrow 0$ . Assume initial configuration  $\{\phi\}$ , and calculate initial  $\{c\}$  such as average intensity value

$$c_l = \frac{\sum_{i \in \mathcal{P}} u_i \delta_{\phi_i, l}}{\sum_{i \in \mathcal{P}} \delta_{\phi_i, l}},$$

where  $\delta_{i,j}$  is Kronecker's delta.

**Step 2** Sample the voxel  $i_0$  on  $\mathcal{P}$  and the trial value  $m \in [1, n]$  for  $\{\phi\}$  from a uniform distribution. Set the trial configuration as

$$\phi'_i = \begin{cases} \phi_i, & i \neq i_0 \\ m, & i = i_0; \end{cases}$$

thus,  $\{\phi'\}$  is the configuration that only the value at voxel  $i_0$  is changed to  $m$  from  $\phi_{i_0}$ .

**Step 3** Calculate the change of energy value  $E(\{c'\}, \{\phi'\}) - E(\{c\}, \{\phi\})$ , which will be described later in this paper via Eq. (8).

**Step 4** Update the configuration  $\{\phi\} \leftarrow \{\phi'\}$  with the probability  $\min(1, \exp[-(E(\{c'\}, \{\phi'\}) - E(\{c\}, \{\phi\}))/T])$ , otherwise the configuration remains unchanged.

**Step 5** Repeat the procedures from Step 2 to Step 4  $N$  times, where  $N$  is the number of voxels in image. Memorize current energy value as  $E_s$  and increment MCS  $s \leftarrow s + 1$ .

**Step 5'** In every  $a_0$  step of  $s$ , let  $T \leftarrow T \times \alpha$  with  $\alpha \in (0, 1)$  as constant.

**Step 6** Repeat the procedures in Step 5 and 5' until the simulation meets the stop condition, which is the state when difference in energy between two steps of lattice is smaller than some small value.

As typical values of  $a_0$  and  $\alpha$ , which control the process of annealing, we will take  $a_0 = 5$  and  $\alpha = 0.98$ , later. By this annealing procedure given in Step 5', the energy decreases. For simulated annealing, the computational time strongly depends on the cost to calculate  $E(\{c'\}, \{\phi'\}) - E(\{c\}, \{\phi\})$  in Step 3. It is not practical if the calculation of order  $N$  is needed for single configuration change process. In the case of Mumford–Shah segmentation model, as Watanabe et al. (2011) showed, only the computation of local energy change around  $\phi_{i_0}$  is needed for calculating a trial energy  $E(\{c'\}, \{\phi'\})$  in Step 3. We write the energy in terms of the variables,

$$\begin{aligned} N_l &= \sum_i \delta_{\phi_i, l}, \\ U_l &= \sum_i u_i \delta_{\phi_i, l}, \\ c_l &= U_l / N_l, \end{aligned} \quad (6)$$

where  $N_l$  means the number of voxels with phase label  $l$  and  $U_l$  means the total intensity of voxels with phase label  $l$ . If  $\phi_{i_0} = k$  at the voxel  $i_0$  is changed to  $\phi'_{i_0} = m$  on Step 2 in Algorithm A, then the new values on trial configuration  $N'_k, N'_m, U'_k, U'_m$  and  $c'_k, c'_m$  are given by,

$$\begin{aligned} N'_k &= N_k - 1, & N'_m &= N_m + 1 \\ U'_k &= U_k - u_{i_0}, & U'_m &= U_m + u_{i_0} \\ c'_k &= U'_k / N'_k, & c'_m &= U'_m / N'_m. \end{aligned} \quad (7)$$

Finally, the energy change in Mumford–Shah functional is calculated as

$$E(\{c'\}, \{\phi'\}) - E(\{c\}, \{\phi\}) = -c'_k U'_k + c_k U_k - c'_m U'_m + c_m U_m + \mu \Delta |C|, \quad (8)$$

where the first to fourth terms are the change of data terms and the last term is the change of regularization term. The energy change of the regularization term can be evaluated as

$$\Delta |C| = \mu \sum_{j \in \mathcal{N}(i_0)} (|m - \phi_j| - |k - \phi_j|). \quad (9)$$

Since Eq. (8) only includes local variables, we can compute the energy difference rapidly. Hereafter, we denote the three-dimensional Monte Carlo method described above as “the original Monte Carlo method”.

As will be described in Section 4, the original Monte Carlo method works effectively in many cases, however, it may converge to local minima in some cases. In order to improve this situation, we introduce an iterative descent process to the original Monte Carlo method. We call it “the modified Monte Carlo method”.

In contrast with the original Monte Carlo method which calculates optimal  $\{\phi\}$  and  $\{c\}$  simultaneously in single simulated annealing procedure, the modified Monte Carlo method operates simulated annealing procedures iteratively; The modified Monte Carlo method calculates optimal  $\{\phi\}$  by simulated annealing under the condition that parameters  $\{c\}$  are fixed, then update the parameters  $\{c\}$  using the obtained  $\{\phi\}$ , and iterates this step until the energy converges. This iterative structure of the modified Monte Carlo method is a similar idea to the graph cuts method by [Bae and Tai \(2008\)](#). In their work, the max-flow algorithm was applied iteratively for minimizing the Mumford–Shah segmentation model. We apply the simulated annealing instead of the max-flow algorithm.

The step-by-step algorithm of the modified Monte Carlo method is described as follows:

#### Algorithm B

**Step 1** Let parameter  $t \leftarrow 0$  as iteration counter of the simulated annealing procedure. Assume initial average intensity parameter,  $\{c\}^t \leftarrow \{c\}^0$ .

**Step 2** By using simulated annealing, calculate optimal configuration  $\{\phi\}^t$  that minimizes the energy under the condition that the parameters  $\{c\}$  are fixed  $\{c\} = \{c\}^t$ ,

$$\{\phi\}^t = \arg \min_{\phi} E(\{c\} = \{c\}^t, \{\phi\}).$$

This step includes Step 2 to 6 in Algorithm A. In the simulated annealing procedure of this step, the way of calculating energy difference is no longer the same as Eq. (8) in Algorithm A because the parameter  $\{c\}$  are fixed to  $\{c\} = \{c\}^t$ . The energy difference is calculated as follows: If  $\phi_{i_0} = k$  at the voxel  $i_0$  is changed to  $\phi'_{i_0} = m$ , the energy change is easily calculated as

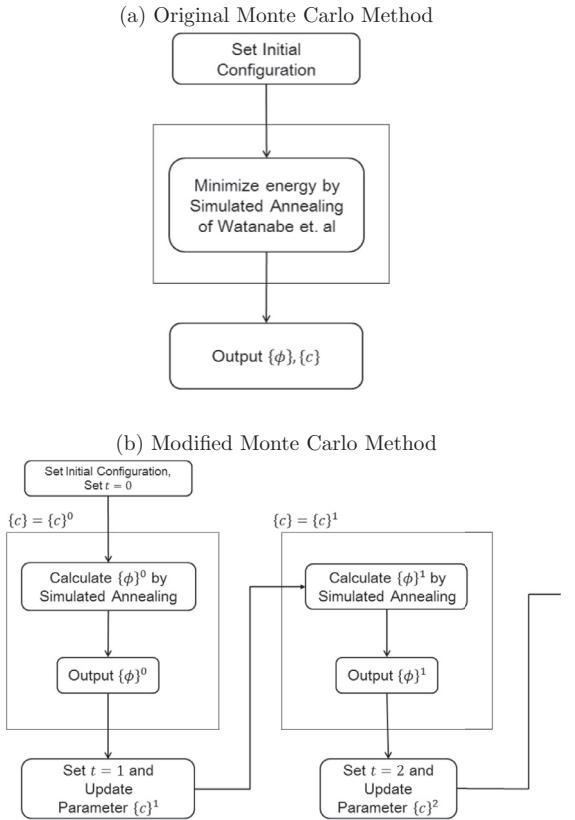
$$\begin{aligned} E(\{c\}, \{\phi'\}) - E(\{c\}, \{\phi\}) \\ = (u_i^0 - c_m)^2 - (u_i^0 - c_k)^2 + \mu \Delta |C| \\ = -2u_i^0(c_m - c_k) + (c_m)^2 - (c_k)^2 + \mu \Delta |C|. \end{aligned} \quad (10)$$

**Step 3** Update parameter  $\{c\}^t \leftarrow \{c\}^{t+1}$  by calculating

$$c_l = \frac{\sum_{i \in \mathcal{P}} u_i^0 \delta_{\phi_l^0, l}}{\sum_{i \in \mathcal{P}} \delta_{\phi_l^0, l}}. \quad (11)$$

**Step 4** Repeat the procedures in Step 2, 3 until the simulation meets stop condition.

[Fig. 1](#) shows the schematic illustration of the original Monte Carlo method (a) and that of the modified Monte Carlo method (b). The figure illustrates that the modified Monte Carlo method applies the simulated annealing iteratively in contrast with the original Monte Carlo method which calculates optimal  $\{\phi\}$  and  $\{c\}$  simultaneously. It must be noted that the modified Monte Carlo method is not ensured to converge to global minimum solution theoretically because of iterative structure. However, in practice it converges to lower energy solution than the original Monte Carlo method, which is shown in [Section 4](#). We here make a comment on the initial configuration. We use the isodata algorithm ([Velasco, 1980](#)) for creating the initial configuration, which follows the method of the graph-cut ([Bae and Tai, 2008](#)). Since the isodata algorithm is a type of thresholding algorithm that only considers the intensity histogram of input image, the output pattern provided by the isodata algorithm is over-segmented, and the energy is higher than the lowest energy solution. However, the configuration created by the isodata algorithm is rather close to the final



**Fig. 1.** Illustration of simulation flow of the original Monte Carlo method (a) and the modified Monte Carlo method (b). The modified Monte Carlo method has one more outer loop than the original Monte Carlo method.

solution compared to the random configuration; thus, this choice of initial configuration gives faster convergence of the Monte Carlo methods.

#### 4. Numerical results

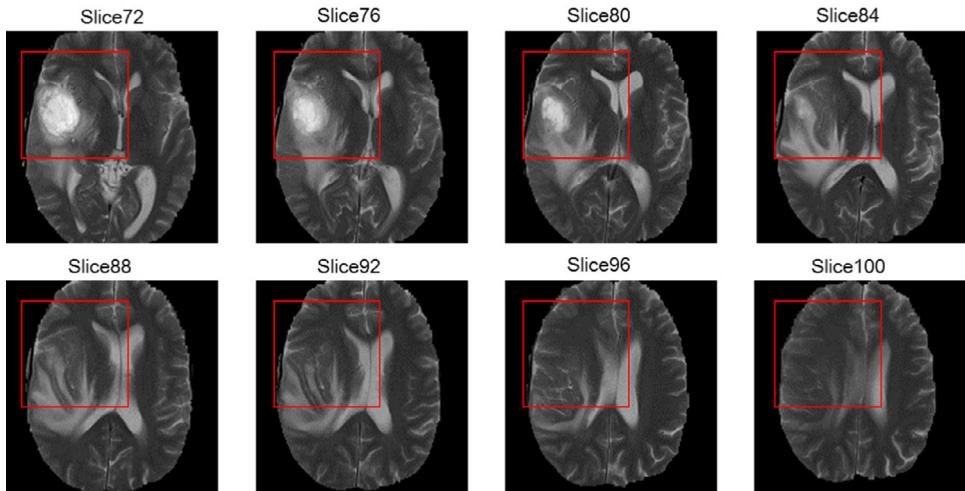
In this section, we show the results of energy minimization methods applied to three-dimensional brain tumor images. We have applied the original Monte Carlo method and the modified Monte Carlo method to testing images. We will show typical results of Monte Carlo simulation from 20 times execution with different random numbers. We have also applied the graph cuts method proposed in ref. [Gurholt and Tai \(2009\)](#) for comparison.

In discussing the energy model, which is given in Eq. (5), there are two parameters, number of phases  $n$  and smoothing parameter  $\mu$ . We fix  $n = 4$  and  $\mu = 1000$  for comparing several methods to find energy minimum solution. In the case of small  $\mu$ , the initial configuration will give the energy-minimum solution. If we increase  $\mu$ , this solution becomes unstable. We chose the case that the searching for the energy-minimum solution becomes difficult, say  $\mu = 1000$ . Later, we will discuss other choices of  $n$  and  $\mu$ .

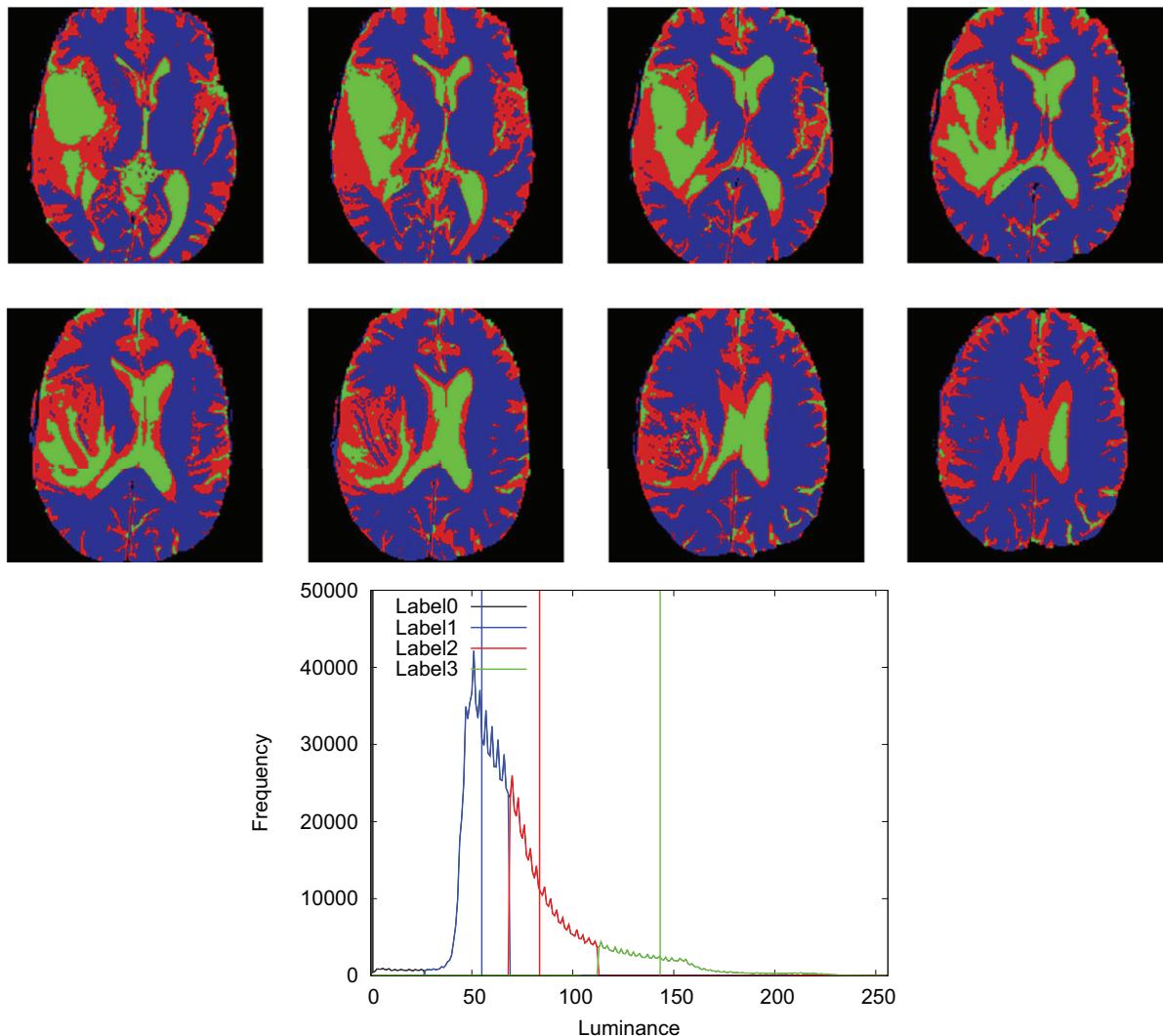
##### 4.1. Testing images

For testing images, we use brain tumor image data. Brain tumor image data used in this work were obtained from the MICCAI 2012 Challenge on Multimodal Brain Tumor Segmentation ([Miccai, 2012](#)) organized by B. Menze, A. Jakab, S. Bauer, M. Reyes, M. Prastawa, and K. Van Leemput. The challenge database contains fully anonymized images from the following institutions: ETH Zurich, University of Bern, University of Debrecen, and University of Utah.

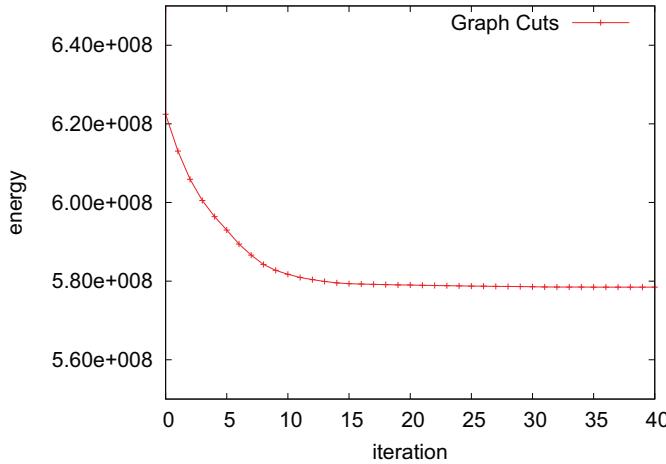
[Fig. 2](#) shows testing images made from challenging database. Images are obtained from T2-enhanced brain images with tumor.



**Fig. 2.** 8-bit multi-slice testing images. Image size is  $(160 \times 160 \times 160)$  voxels. Red square indicates smaller testing images obtained by trimming the part of lesion. Image size is  $(80 \times 80 \times 80)$  voxels. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 3.** The pattern and intensity histogram obtained by isodata algorithm applied to Fig. 2. Colors on each figure correspond to phase labels: 0 (black), 1 (blue), 2 (red), 3 (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 4.** The process of energy convergence of graph cuts method.

Image size is adjusted to  $(160 \times 160 \times 160)$  voxels). Image depth is converted from 16-bit to 8-bit. Slice numbers are 72, 76, 80, ..., 100. In order to see the size dependence of memory consumption and computational time, we also use smaller image indicated by red rectangle in Fig. 2. It was obtained by trimming the part of lesion. Image size is  $(80 \times 80 \times 80)$  voxels. This smaller image will be used later in the discussion of Section 4.7 (Table 2).

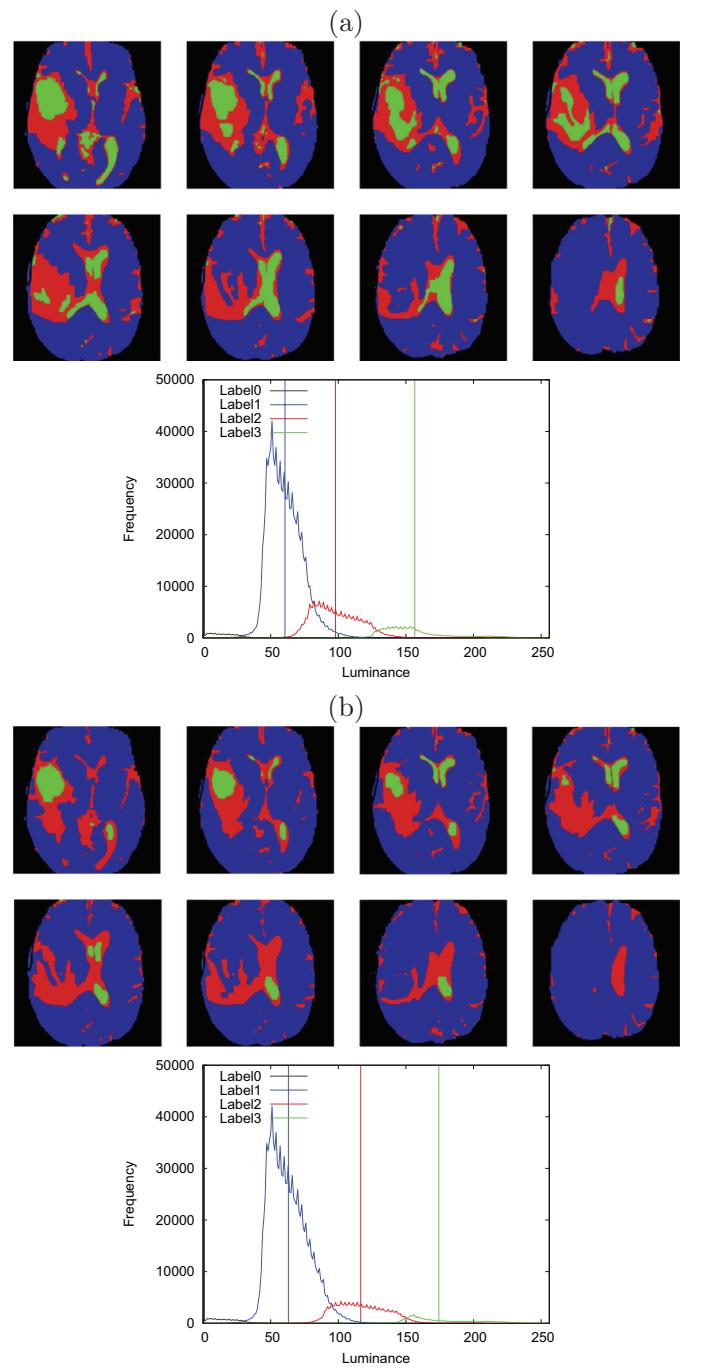
As explained in Section 3, we use the isodata algorithm for creating the initial configuration. Fig. 3 gives the initial four-segmented pattern of testing image given in Fig. 2. The histogram of intensities for segmented regions is also shown in Fig. 3 by different colors. The distribution of intensities is completely separated because of thresholding algorithm of isodata. It means that the pattern of isodata algorithm does not include the spatial correlation given in the regularization term in Eq. (5).

#### 4.2. Results of the graph cuts method

We first show the results of the graph cuts methods of Gurholt and Tai (2009). As a max-flow algorithm, which is used for minimizing the energy, we implement that by Boykov and Kolmogorov (2004). Their method has iterative structure: Computes optimal  $\{\phi\}$  by a max-flow algorithm under the condition of  $\{c\}$  fixed, then update  $\{c\}$  by given  $\{\phi\}$ . Since max-flow algorithm can find optimal solution on each iteration, this method is ensured to decrease energy on each step.

Here, we examined the relation between energy-convergence and iteration step. As simulation parameters we took  $n = 4$  and  $\mu = 1000$ . Fig. 4 shows the process of convergence of the graph cuts method. It can be seen that the energy monotonously decreases with iteration step. Fig. 5(a) and (b) show example of output images at iteration number 5 (a) and 40 (b), respectively. The histograms of intensity-distribution are also shown. By comparing Fig. 4 and the two images in Fig. 5, we can see the relation between the energy value and the output. The differences in energy are reflected on the differences in pattern and intensity-histogram. The output pattern and intensity-distribution for Fig. 5(b), which are different from those given in Fig. 5(a), clearly show the lesion part. It reflects upon the energy convergence process in Fig. 4.

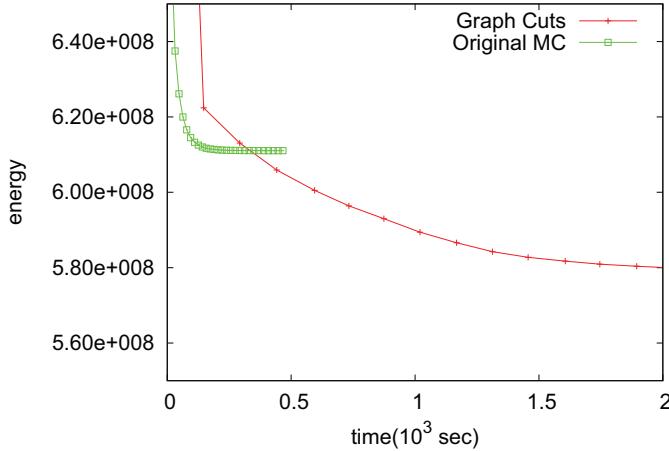
In discussing the performance of the original Monte Carlo method and the modified Monte Carlo method, we will examine not only the convergence of energy but also the output pattern and the histogram.



**Fig. 5.** The output patterns of the graph cuts method applied to Fig. 2 at iteration No. 5(a) and iteration No. 40(b). The histogram of the intensity, luminance, for each phase  $0 \sim 3$  is also given. Colors on each figure correspond to phase labels: 0 (black), 1 (blue), 2 (red), 3 (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

#### 4.3. Results of original Monte Carlo method

In this subsection, we show the result of the original Monte Carlo method, which is a process of the simulated annealing, applied to the three-dimensional image of Fig. 2. In the original Monte Carlo method, which was shown in Algorithm A on Section 3, there are three simulation parameters; initial pseudo-temperature  $T^0$ , annealing interval parameter  $a_0$  and annealing rate  $\alpha$ . We chose  $T^0 = 3000$ ,  $a_0 = 5$  and  $\alpha = 0.98$ , as a typical value of the annealing schedule. We chose the schedule such that we



**Fig. 6.** The process of energy convergence of original Monte Carlo method. The energy is plotted as a function of real computational time to compare with the graph cuts method (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

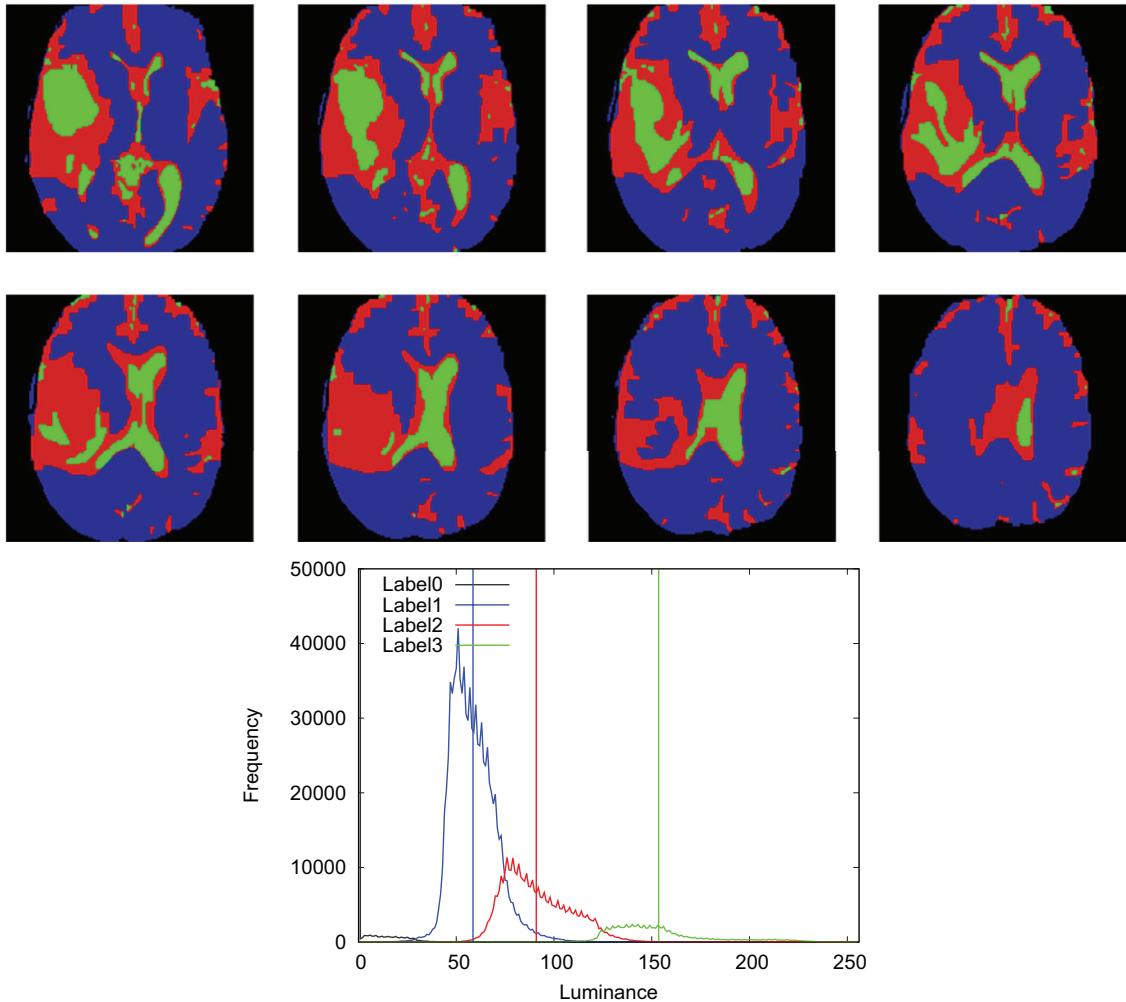
decrease the temperature neither too quickly nor too slowly. For the choice of the initial temperature, we make the first and second terms of the right-hand side of Eq. (5) the same order. We should note that  $u_i$  takes values between 0 and 255.

In Fig. 6, we show the process of energy convergence using the original Monte Carlo method applied to the Mumford–Shah segmentation model. To compare with the graph cuts method, we plot the energy as a function of real computational time instead of the Monte Carlo steps using the same machine, Intel Core i7-3630QM (2.4 GHz, 6 MB Cache), 8 GB RAM. The energy convergence of the graph cuts method is shown by the red line in the graph. In Fig. 7, we show the segmented pattern of final output and the histogram of intensity-distribution. From Fig. 6, we see that the original Monte Carlo method is trapped at a local minimum with higher energy, and it does not reach the low energy solution obtained by the graph cuts method. We can confirm this trapping also from the output pattern in Fig. 7, which is close to the intermediate pattern of Fig. 5(a) by the graph cuts method, and different from the final pattern of Fig. 5(b). We also observe this behavior from the intensity histogram of Figs. 5 and 7.

To conclude this subsection, we find that the original Monte Carlo method is fast for computational convergence, but the accuracy of energy minimization is not good for some cases.

#### 4.4. Results of modified Monte Carlo method

In this subsection, we show the result of the modified Monte Carlo method, which was proposed to improve the problem of convergence accuracy of the original Monte Carlo method. In the

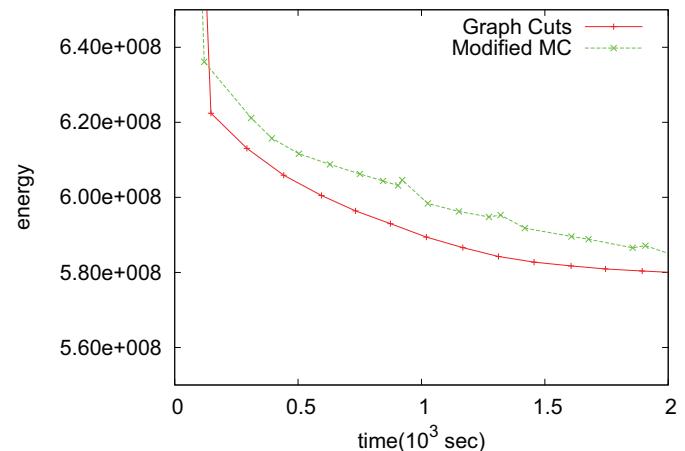


**Fig. 7.** The final output patterns and histogram of the original Monte Carlo method applied to Fig. 2. The histogram of the intensity, luminance, for each phase 0 ~ 3 is also given. Colors on each figure correspond to phase labels: 0 (black), 1 (blue), 2 (red), 3 (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

modified Monte Carlo method, the energy is minimized to search for the optimal  $\{\phi\}$  by the Monte Carlo method on the condition that  $\{c\}$  are fixed; then,  $\{c\}$  are updated by the obtained  $\{\phi\}$ . We make a total energy minimization by iterating this updating process.

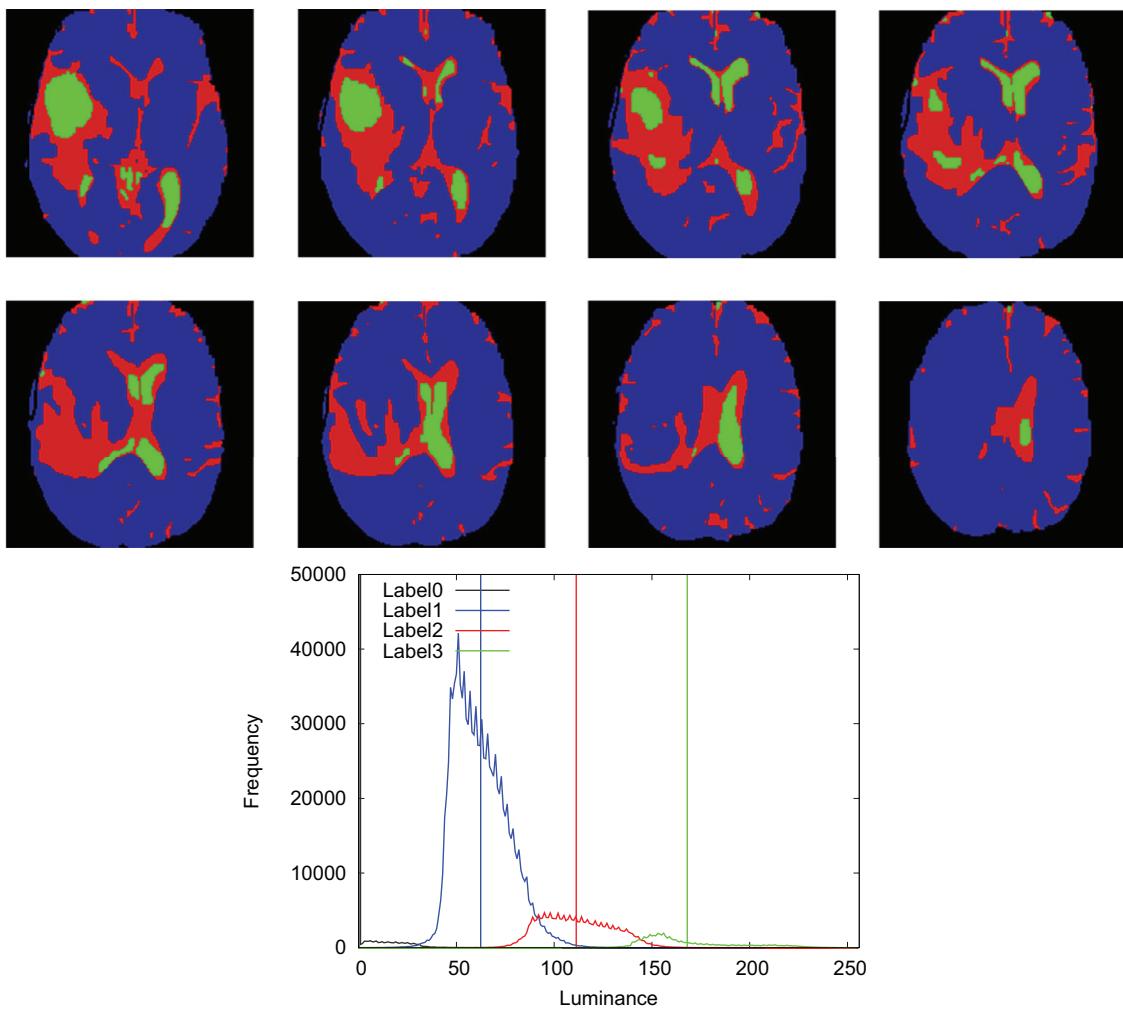
The max-flow algorithm in the graph cuts method of Gurholt and Tai (2009) is replaced by the Monte Carlo simulated annealing. More generally, it is based on the strategy to make convex optimization (Boyd and Vandenberghe, 2004) iteratively in order to solve a non-convex optimization problem (Gasso et al., 2009). The simulation parameters for the Monte Carlo method are the same as those for the original Monte Carlo method;  $T^0 = 3000$ ,  $a_0 = 5$  and  $\alpha = 0.98$ .

In Fig. 8, the energy-convergence process of the modified Monte Carlo method is shown. We again chose the real simulation time as a horizontal axis in order to compare with other methods. The result of the graph cuts method is shown by the red line. From Fig. 8, we see that the problem of energy trap to a higher energy solution was greatly improved in the case of the modified Monte Carlo method because the modified Monte Carlo method has one more outer loop than the original Monte Carlo method; the modified Monte Carlo method reaches a lower energy solution of the final output of the graph cuts method. We can also observe this behavior from the output pattern shown in Fig. 9, which is close to the graph cuts pattern shown in Fig. 5. We may conclude that the performance of the energy minimization of the modified Monte



**Fig. 8.** The process of energy convergence of modified Monte Carlo method. Each point of the modified Monte Carlo method includes the whole process of the simulated annealing for one iteration. The energy is plotted as a function of real computational time to compare with the graph cuts method (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Carlo method is the same as that of the graph cuts method, although small thermal fluctuations remain. On the other hand, the computational time of the modified Monte Carlo method is longer



**Fig. 9.** The output patterns and histogram of result of the modified Monte Carlo method applied to Fig. 2 at iteration No. 40 (final). Colors on each figure correspond to phase labels: 0 (black), 1 (blue), 2 (red), 3 (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

than that of the original Monte Carlo method because the modified Monte Carlo method is an iterating process of simulated annealing.

#### 4.5. Results of modified Monte Carlo method with block-spin transformation

In order to conquer the problem of long computational time for the modified Monte Carlo method, we here propose to introduce block-spin transformation process.

To accelerate the convergence, Watanabe et al. already proposed to use the idea of the block-spin transformation in the original Monte Carlo method (Watanabe et al., 2011). By starting with the coarse-grained system instead of the original lattice, the convergence of simulation can be accelerated drastically. For three dimensions, each operation of block-spin transformation reduces the number of voxels by eightfold. Block-spin transformation is performed by taking eight neighboring voxels and replacing them by one voxel with the average intensity of them. We iterate this process in the Monte Carlo optimization. If we start with the coarse-grained system with  $b$  times, the time for one Monte Carlo step (MCS) per spin is roughly  $1/8^b$  of that for the original size. The invariance of the energy under block-spin transformation was given by Law et al. (2008); the energy of regularization term is invariant by setting the parameter  $\mu$  as  $\mu \rightarrow \mu/2$  in the block-spin transformation. This statement holds both for the case of regularization energy term of isotropic Potts interaction, Eq. (3) and  $TV_1$  interaction, Eq. (4).

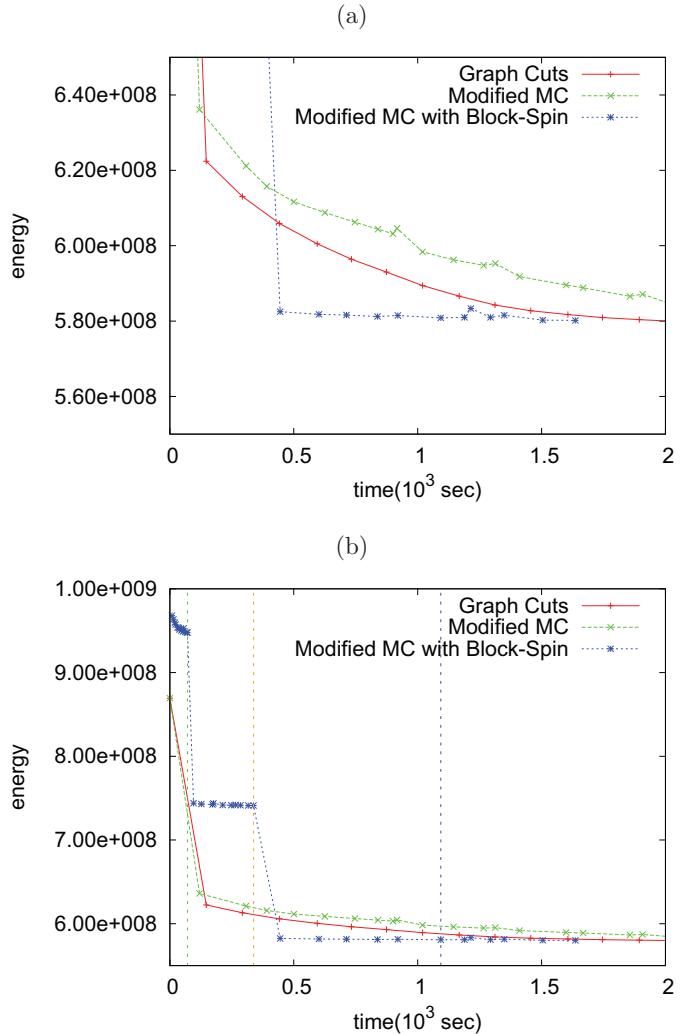
We show the energy convergence process of the block-spin modified Monte Carlo method as a function of computational time in Fig. 10. We also plot the data for the graph cuts method and the modified Monte Carlo method for comparison. From Fig. 10, we find that the block-spin modified Monte Carlo method reaches the lowest energy solution with a shortest time compared to other methods. The obtained value of energy is almost the same as the solution by the graph cuts method.

We show the pattern of block-spin Monte Carlo method with block-spin levels 2, 1 and 0 in Figs. 11, 12 and 13, respectively. The time at which we took each pattern is indicated by green, orange, and blue vertical dotted lines in Fig. 10 (b). The pattern and histogram shown in Fig. 11 are those for block-spin level 2 of system-size  $40 \times 40$ . Although the pattern and histogram of the final output of the modified Monte Carlo method shown in Fig. 7 are close to the higher energy pattern shown in Fig. 5(a) in the graph cuts method, the pattern of Fig. 11 is an intermediate pattern between Fig. 5(a) and (b). That is, the state given in Fig. 11 captures characteristic pattern of Fig. 5(b) in very short time, although the value of energy is higher because of coarse-graining. Fig. 12 is the pattern and histogram of block-spin level 1 with system-size  $80 \times 80$ , and the pattern and histogram are very close to the final output of the graph cuts method, Fig. 5(b). Finally, in Fig. 13 of block-spin level 0, we obtained the desired solution of the graph cuts method in the pattern, the intensity-histogram, and the value of energy.

To conclude, in the block-spin modified Monte Carlo method, we can search for characteristic pattern of final output solution in short time by making simulation for reduced size systems of block-spin transformation. We can reach the final solution with very small steps compared to the modified Monte Carlo method. We have confirmed that the drawback of the modified Monte Carlo method that it takes long time has been conquered.

#### 4.6. Results of the graph cuts method with block-spin transformation

In the previous subsection, we have shown that the modified Monte Carlo Method with the block-spin is very effective. We now consider the possibility of applying the block-spin transformation to the graph-cuts method.

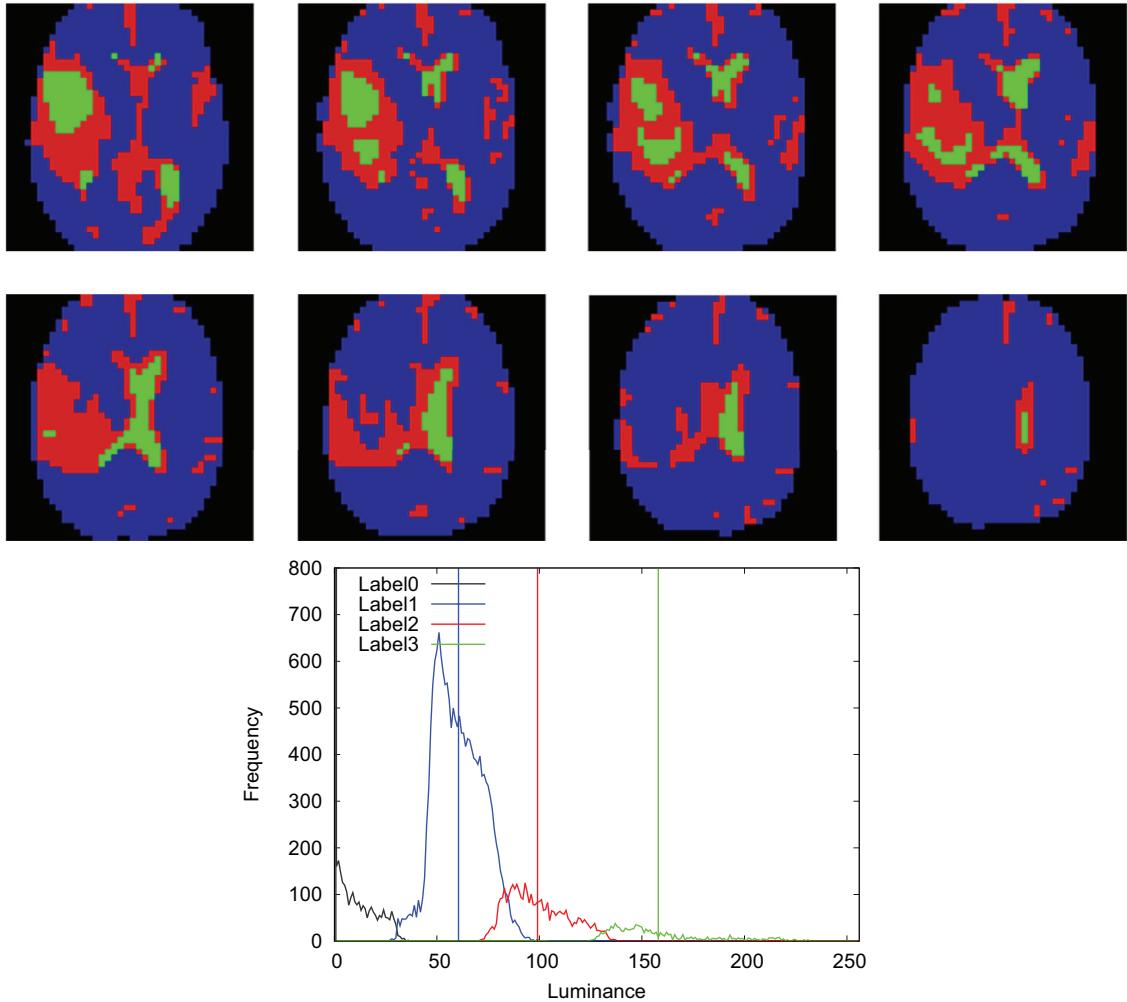


**Fig. 10.** (a) The process of energy convergence of block-spin modified Monte Carlo method as a function of real computational time. The result of the graph cuts method and the modified Monte Carlo method (without block-spin) are also shown. (b) The same plot as (a) but the vertical axis is enlarged in order to show the step-wise energy decrease process of block-spin systems explicitly.

We perform the block-spin transformation for the original image, and search for the minimum energy segmented output of small size with the max-flow algorithm. It takes short time for this process because the system size is small. Then, we use the enlarged output of the previous step as the input of the max-flow algorithm of the next step. We show the process of convergence of the graph-cuts method with the block-spin transformation in Fig. 14. The process for the regular graph-cuts method, which was shown in Fig. 4, is also plotted there. It is to be noted that the scale of the vertical axis is different. Since the input of the final process is already close to the minimum energy solution of the final size, smaller steps are needed for iteration. As a result, we can save the computational time of the graph-cuts method with the introduction of the block-spin transformation.

#### 4.7. Effects of the system size and the number of phases

We have shown a good performance of the block-spin modified Monte Carlo Method as an energy minimization tool for the Mumford-Shah segmentation model. As a testing image, we picked up the three-dimensional brain image with tumor of size  $160 \times 160 \times 160$  voxels and set the number of segmented phases  $n$  as 4. Here, we discuss the effects of the size and the number of phases



**Fig. 11.** The output patterns and histogram of the modified Monte Carlo method applied to Fig. 2 at block-spin level 2, which is indicated by green vertical dotted line on Fig. 10(b). Colors on each figure correspond to phase labels: 0 (black), 1 (blue), 2 (red), 3 (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

*n*. We pay attention to the memory size, the computational time and the obtained energy by various methods of energy minimization.

In Table 1, we compare the memory allocation, the computational time, and the obtained energy by the graph cuts method, the graph cuts method with block-spin transformation (BSGC), the original Monte Carlo (MC) method, the modified Monte Carlo (MMC) method and the block-spin modified Monte Carlo (BSMMC) method. The testing image is that given in Fig. 2 with size of  $160 \times 160 \times 160$  voxels, and we compare the cases of  $n = 4, 3$  and 2. In Table 2, we tabulate the data for the smaller image with size of  $80 \times 80 \times 80$  voxels shown by red rectangle in Fig. 2.

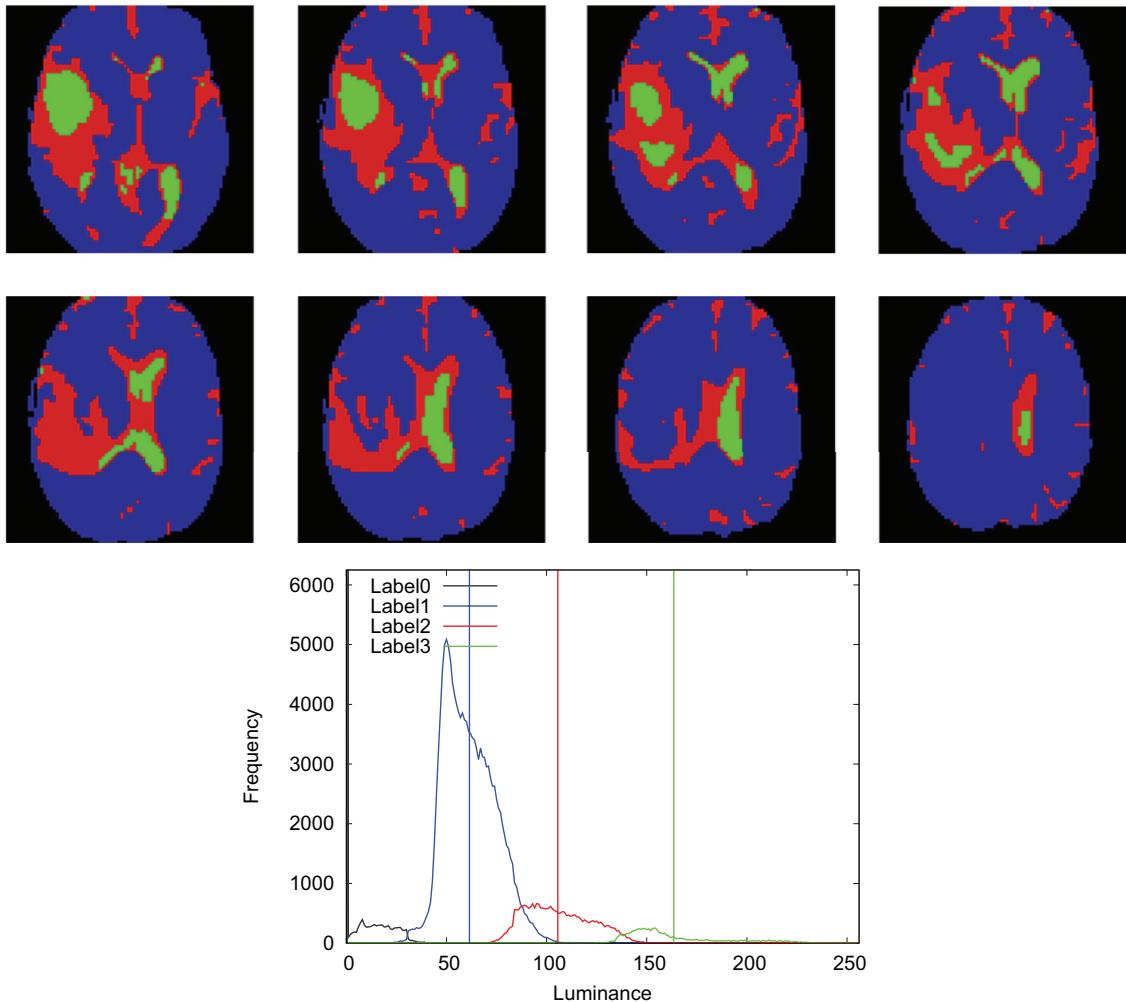
The Monte Carlo methods are superior to the graph cuts method in terms of memory consumption. In the Monte Carlo methods, the necessary quantities of memory allocation essentially depend on the number of voxels  $N$  and do not depend on the number of the segmentation  $n$ . Moreover, even if we employ the block-spin Monte Carlo methods, only a small increase in memory allocation is required because the size of block-spin systems is  $1/8^k$  ( $k$  being the block-spin level) times smaller than the original system. On the other hand, in the graph cuts method, the variables are allocated to the interactions between voxels; as a result, the memories proportional to  $(6 \times N \times n)$  are required. Thus, especially for the multi-phase segmentation problem, the superiority in memory consumption of the Monte Carlo methods over the graph cuts

**Table 1**

Comparison of memory allocation, computational time and energy between the graph cuts method, the graph cuts method with block-spin transformation (BSGC), the original Monte Carlo method, the modified Monte Carlo method (MMC), and the block-spin modified Monte Carlo method (BSMMC). Testing image is Fig. 2 ( $160 \times 160 \times 160$ ),  $\mu = 1000$ , number of regions  $n = 4, 3, 2$  case.

Method	n	Memory (MByte)	Time(sec)	Energy
Graph cuts	4	3499.6	5977	$5.785 \times 10^8$
BSGC	4	3496.9	4424	$5.785 \times 10^8$
Original MC	4	47.2	467	$6.110 \times 10^8$
MMC	4	47.2	4329	$5.800 \times 10^8$
BSMMC	4	54	1637	$5.801 \times 10^8$
Graph cuts	3	2314.1	1621	$6.575 \times 10^8$
BSGC	3	2318.6	1177	$6.575 \times 10^8$
Original MC	3	47.2	434	$6.577 \times 10^8$
MMC	3	47.2	728	$6.577 \times 10^8$
BSMMC	3	54	301	$6.579 \times 10^8$
Graph cuts	2	1134.3	4	$1.379 \times 10^9$
BSGC	2	1135.7	6	$1.379 \times 10^9$
Original MC	2	47.2	273	$1.379 \times 10^9$
MMC	2	47.2	45	$1.379 \times 10^9$
BSMMC	2	54	9	$1.386 \times 10^9$

method is noticeable. We here make a comment on the graph cuts method with block-spin transformation. From Tables 1 and 2, we find that BSGC can save the computational time by introducing a



**Fig. 12.** The output patterns and histogram of the modified Monte Carlo method applied to Fig. 2 at block-spin level 1, which is indicated by orange vertical dotted line on Fig. 10(b). Colors on each figure correspond to phase labels: 0 (black), 1 (blue), 2 (red), 3 (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 2**

Comparison of memory allocation, computational time and energy between the graph cuts method, the graph cuts method with block-spin transformation (BSGC), the original Monte Carlo method, the modified Monte Carlo method (MMC), and the block-spin modified Monte Carlo method (BSMMC). Testing image is Fig. 2 ( $80 \times 80 \times 80$ ),  $\mu = 1000$ , number of regions  $n = 4, 3, 2$  case.

Method	n	Memory (MByte)	Time(sec)	Energy
Graph cuts	4	437.4	50	$1.846 \times 10^8$
BSGC	4	437.7	24	$1.846 \times 10^8$
Original MC	4	3.4	59	$1.899 \times 10^8$
MMC	4	3.4	123	$1.887 \times 10^8$
BSMMC	4	3.7	70	$1.893 \times 10^8$
Graph cuts	3	290.2	17	$2.454 \times 10^8$
BSGC	3	290.5	13	$2.454 \times 10^8$
Original MC	3	3.4	52	$2.454 \times 10^8$
MMC	3	3.4	45	$2.454 \times 10^8$
BSMMC	3	3.7	22	$2.456 \times 10^8$
Graph cuts	2	142.9	5	$5.401 \times 10^8$
BSGC	2	143.1	3	$5.401 \times 10^8$
Original MC	2	3.4	42	$5.401 \times 10^8$
MMC	2	3.4	49	$5.403 \times 10^8$
BSMMC	2	3.7	37	$5.402 \times 10^8$

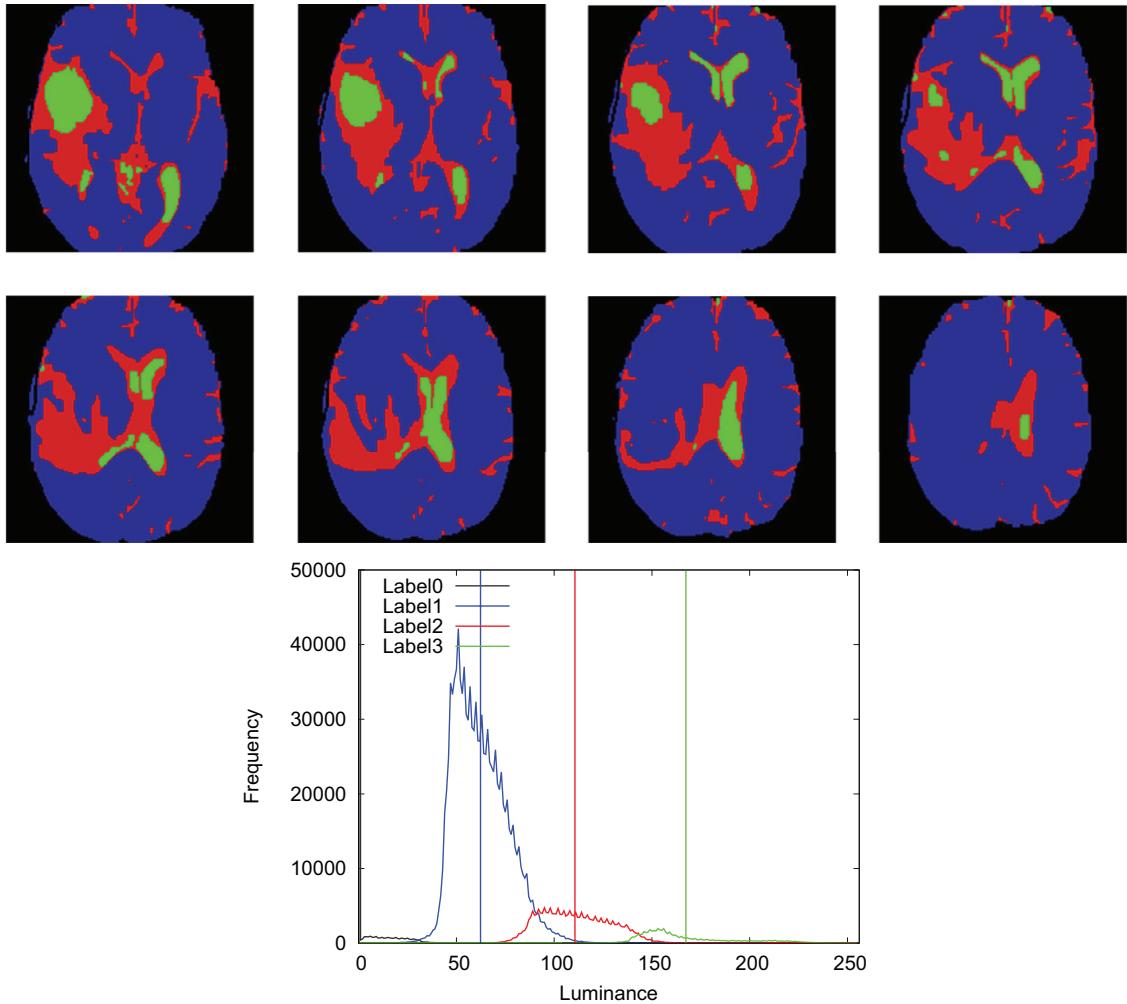
simple procedure of block-spin transformation, and reach the same required lowest energy. But the memory consumptions of BSGC remain the same as those of the original graph cuts.

The Monte Carlo methods also search for the lower energy states very close to the graph cuts solution in a reasonably short time. The minimum energy search by the original Monte Carlo method is trapped at a local minimum in some cases for larger sizes and larger  $n$ , but the modified Monte Carlo methods conquer this drawback.

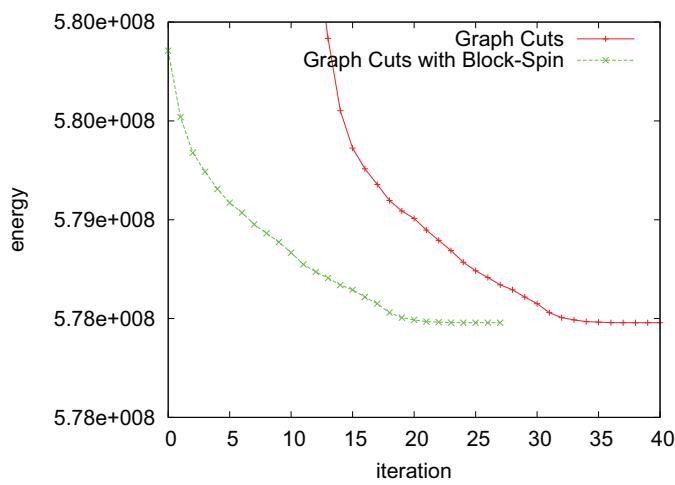
When we apply the simulated annealing, the computational time depends on the stop condition or the annealing schedule. Especially, in the case of the block-spin modified Monte method we propose in this paper, there is a choice of turning time of block-spin levels. In the data shown in Tables 1 and 2, we regard that the block-spin systems reach the equilibrium if the energy change in Step 4 of Algorithm B is smaller than  $1.0 \times 10^{-4}$ , or the obtained energy is greater than that of the previous step.

We mention the effects of the smoothing parameter  $\mu$ . In the case of smaller  $\mu$ , the solution of the isodata algorithm is enough. For larger  $\mu$ , the energy minimization problem becomes complex and the original Monte Carlo method sometimes converges to local minima; the block-spin modified Monte Carlo method works well for this situation.

We here make a comment on the computational time of the graph cuts method. The computational time shown in Tables 1 and 2 are the time when a simulation converges to “lowest” minimum solution. Practically, we can break computation when a simulation converges to sufficiently lower energy. For comparison of



**Fig. 13.** The output patterns and histogram of the modified Monte Carlo method applied to Fig. 2 at block-spin level 0 (= original size), which is indicated by blue vertical dotted line on Fig. 10(b). Colors on each figure correspond to phase labels: 0 (black), 1 (blue), 2 (red), 3 (green). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



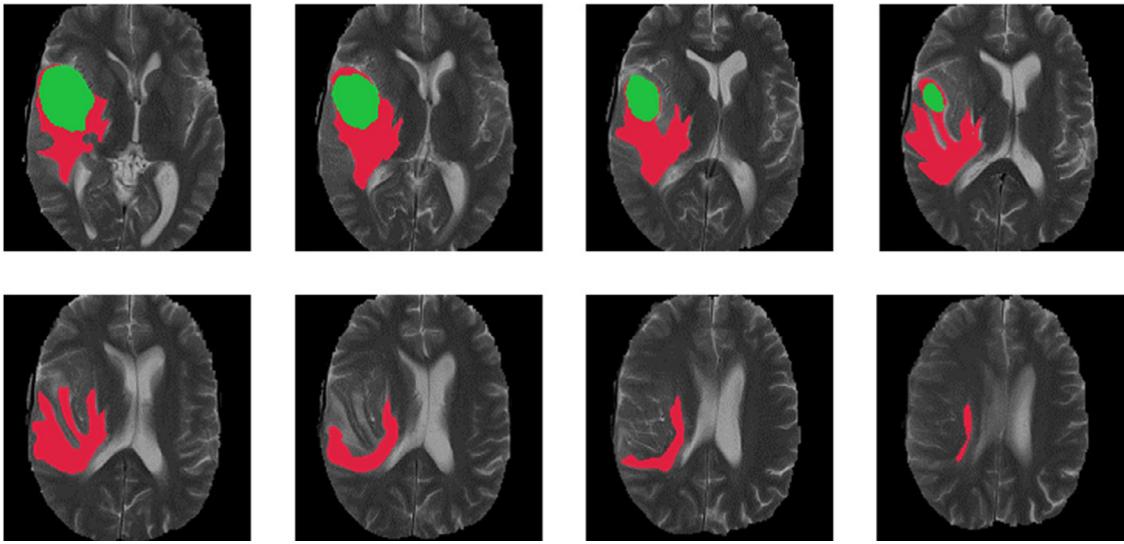
**Fig. 14.** The comparison between the energy convergence of graph cuts method and that of the graph cuts method with block-spin transformation.

BSMMC, the computational time of the graph cuts method is 2043 sec when the simulation converges to almost the same energy value with the solution of BSMMC, and that of the graph cuts with

block-spin transformation is 330 sec in Table 1 ( $160 \times 160 \times 160$ )  $n = 4$  case. From this observation, we regard the computational time of the BSMC is about the same order with the graph cuts method. It should be noted that we have not made comparison with faster implementations of graph cuts method such as (Juan and Boykov, 2006; Kohli and Torr, 2007; Lombaert et al., 2005). The active graph cuts method is reported to be 2–6 times faster than the method of Boykov and Kolmogorov (2004); the multilevel banded graph cuts method (Lombaert et al., 2005) is reported to be 8 times faster and 8 times lesser memory consumption than the previous interactive segmentation (Boykov and Jolly, 2001); and the dynamic graph cuts method (Kohli and Torr, 2007) is reported to be 5 times faster than the method of Boykov and Kolmogorov (2004) in larger graph.

#### 4.8. Comparison with experts' annotation

In the present paper, we have used the three-dimensional testing image provided by MICCAI challenging database. As well as testing images, MICCAI challenging database also provides the information on lesion parts annotated by clinical experts. Fig. 15 shows the lesion parts of testing image, which was shown in Fig. 2, inferred from experts' annotation; green-colored region denotes “core” region of tumor and red-colored region denotes “edema” region. For evaluating segmentation results, Dice score, sensitiv-



**Fig. 15.** Lesion parts of the testing image (Fig. 2) inferred from the annotations of consensus segmentation by experts provided by MICCAI challenging database (Miccai, 2012). Green-colored region denotes “core” region of tumor and red-colored region denotes “edema” region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 3**

Dice score, sensitivity and specificity of each tumor region calculated from the result of the modified Monte Carlo method with block-spin transformation.

Region	Dice (%)	Sens (%)	Spec (%)
Core	58.8	60.9	99.3
Edema	34.8	86.7	95.6
Whole (core & edema)	53.0	92.0	95.8

ity (true positive rate) and specificity (true negative rate) (Menze et al., 2014) are commonly used. If we obtain a binary map with algorithmic prediction  $P \in \{0, 1\}$  and experts’ annotation  $T \in \{0, 1\}$  for each of tumor regions, Dice score, sensitivity and specificity can be calculated as

$$\text{Dice}(P, T) = \frac{|P_1 \wedge T_1|}{(|P_1| + |T_1|)/2},$$

$$\text{Sens}(P, T) = \frac{|P_1 \wedge T_1|}{|T_1|}, \quad \text{Spec}(P, T) = \frac{|P_0 \wedge T_0|}{|T_0|},$$

where  $\wedge$  is the logical AND operator,  $|\cdot|$  is the size of the set (i.e., the number of voxels belonging to it), and  $P_i$  and  $T_i$  represent the set of voxels where  $P = i$  and  $T = i$  ( $i = 0$  or  $1$ ), respectively. These scores measure voxel-wise overlap of segmented regions.

We calculate these scores by the final output of modified Monte Carlo method with block-spin transformation shown in Fig. 13. For defining binary mapping  $P$  of our results, we regard phase label 3 (green-colored region) as tumor core and phase label 2 (red-colored region) as edema. Table 3 shows the values of Dice score, sensitivity and specificity of each tumor region from the result of the modified Monte Carlo method with block-spin transformation. We also calculated scores of “whole” region that includes tumor core and edema (Menze et al., 2014). The Dice score of the core region, 58.8%, is compatible to the scores of other algorithms reported in Menze et al. (2014), although the average value for various testing images should be compared. It suggests that the Mumford–Shah segmentation model is suitable for multi-phase image segmentation of three-dimensional medical images although it does not consider clinical information.

## 5. Discussions and conclusion

In this research we have applied the Monte Carlo method of simulated annealing (Watanabe et al., 2011) for solving Mumford–Shah segmentation model to three-dimensional image segmentation problem. We have shown that this original Monte Carlo Method works effectively in many cases. It is compatible with the graph cuts method (Gurholt and Tai, 2009); the obtained energy is very close to the solution of the graph cuts with small thermal fluctuations, and the computational time is of the same order with the graph cuts method which was considered to be superior to the previous gradient-descent methods (Bae and Tai, 2008; 2009). However, this original Monte Carlo method converges to local minima in some cases.

In order to conquer this problem, we have introduced an iterative descent process to the original Monte Carlo Method, which we call the modified Monte Carlo Method. In the modified Monte Carlo Method, the energy is minimized to search for the optimal  $\{\phi\}$  by the simulated annealing on the condition that  $\{c\}$  are fixed; then,  $\{c\}$  are updated by the obtained  $\{\phi\}$ . We have overcome the problem of the original Monte Carlo method that converges to local minima, and we can save the computational time by more than 50% by introducing the block-spin transformation procedure. In comparison with the graph cuts method (Gurholt and Tai, 2009), the proposed method has advantages in memory consumption, especially when the number of phases is larger than two, that is, multi-phase segmentation. We have also introduced the idea of the block-spin transformation in the graph cuts method. We have shown that we can save the computational time, but the problem of the memory consumption still remains in the graph cut method. We calculated Dice score, sensitivity and specificity score of the result of our method, to compare with experts’ annotation, provided by MICCAI challenging database. These scores suggest that the Mumford–Shah segmentation model is suitable for multi-phase image segmentation of three-dimensional biomedical images.

To conclude, we have shown an efficient method to solve the multi-phase Mumford–Shah segmentation model for three dimensional images, the modified Monte Carlo method. Because of the advantage of small memory consumption, lower-energy convergence and reasonably short computational time, the modified Monte Carlo method with the block-spin transformation can be ap-

plied to a wide range of three-dimensional images. We are now applying this method to the image segmentation problem of various three-dimensional images, especially biomedical images.

## Acknowledgment

This work was supported by a Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science.

## References

- Bae, E., Tai, X., 2008. Graph cuts for the multiphase Mumford-Shah model using piecewise constant level set methods. *UCLA CAM Rep.* 8, 36.
- Bae, E., Tai, X., 2009. Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation. *Lect. Notes Comput. Sc.* 5567, 1–13.
- Barbu, A., Zhu, S.-C., 2005. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *Pattern Anal. Mach. Intell. IEEE Trans.* 27 (8), 1239–1253.
- Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge university press, Cambridge, UK.
- Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 1124–1137.
- Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 1222–1239.
- Boykov, Y.Y., Jolly, M.-P., 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: Computer Vision, 2001 ICCV 2001. Proceedings. Eighth IEEE International Conference on, Vol. 1. IEEE, pp. 105–112.
- Chan, T., Vese, L., 2001. Active contours without edges. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 266–277.
- Darbon, J., 2007. A note on the discrete binary Mumford-Shah model. *Lect. Notes in Comput. Sci.* 4418, 283–294.
- Darbon, J., Sigelle, M., 2006. Image restoration with discrete constrained total variation part II: levelable functions, convex priors and non-convex cases. *J. Math. Imaging Vis.* 26, 277–291.
- Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S., 2003. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Mach. Learn.* 50 (1–2), 45–71.
- El-Zehiry, N., Elmaghhraby, A., 2007. Brain MRI tissue classification using graph cut optimization of the mumford-shah functional. In: Proceedings of the International Vision and Computing Conference, Hamilton, New Zealand, pp. 321–326.
- El-Zehiry, N., Xu, S., Sahoo, P., Elmaghhraby, A., 2007. Graph cut optimization for the Mumford-Shah model. In: The Seventh IASTED International Conference on Visualization, Imaging and Image Processing. ACTA Press, pp. 182–187.
- Erdogan, C., Paluri, M., Dellaert, F., 2012. Planar segmentation of RGBD images using fast linear fitting and Markov chain Monte Carlo. In: Computer and Robot Vision (CRV), 2012 Ninth Conference on. IEEE, pp. 32–39.
- Gasso, G., Rakotomamonjy, A., Canu, S., 2009. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *Signal Process. IEEE Trans.* 57 (12), 4686–4698.
- Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 721–741.
- Green, P.J., 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82 (4), 711–732.
- Greig, D., Porteous, B., Seheult, A., 1989. Exact maximum a posteriori estimation for binary images. *J. R. Stat. Soc. B Met.* 51, 271–279.
- Gurholt, T., Tai, X., 2009. 3D multiphase piecewise constant level set method based on graph cut minimization. *Numer. Math. Theor. Meth. Appl.* 2 (4), 403–420.
- Ishikawa, H., 2003. Exact optimization for Markov random fields with convex priors. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1333–1336.
- Ishikawa, H., Geiger, D., 1998. Segmentation by grouping junctions. In: IEEE, pp. 125–131.
- Juan, O., Boykov, Y., 2006. Active graph cuts. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 1. IEEE, pp. 1023–1029.
- Kohli, P., Torr, P.H., 2007. Dynamic graph cuts for efficient inference in Markov random fields. *Pattern Anal. Mach. Intell. IEEE Trans.* 29 (12), 2079–2088.
- Kolmogorov, V., Zabih, R., 2004. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 147–159.
- Law, Y., Lee, H., Yip, A., 2008. A multiresolution stochastic level set method for Mumford-Shah image segmentation. *IEEE Trans. Image Process.* 17, 2289–2300.
- Lie, J., Lysaker, M., Tai, X., 2005. Piecewise constant level set methods and image segmentation. *Lect. Notes Comput. Sc.* 3459, 573–584.
- Lie, J., Lysaker, M., Tai, X., 2006. A binary level set model and some applications to Mumford-Shah image segmentation. *IEEE Trans. Image Process.* 15, 1171–1181.
- Lie, J., Lysaker, M., Tai, X., 2006. A variant of the level set method and applications to image segmentation. *Math. Comput.* 75, 1155–1174.
- Lombaert, H., Sun, Y., Grady, L., Xu, C., 2005. A multilevel banded graph cuts method for fast image segmentation. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 1. IEEE, pp. 259–265.
- Menze, B., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al., 2015. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging* 34, 1993–2024.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21, 1087–1091.
- Miccai, 2012. Challenge on Multimodal Brain Tumor Segmentation. <http://www.imm.dtu.dk/projects/BRATS2012>.
- Mumford, D., Shah, J., 1989. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* 42, 577–685.
- Osher, S., Fedkiw, R., 2003. Level Set Methods and Dynamic Implicit Surfaces. Springer Verlag.
- Sashida, S., Okabe, Y., Lee, H.K., 2014. Comparison of multi-label graph cuts method and monte carlo simulation with block-spin transformation for the piecewise constant Mumford-Shah segmentation model. *Comput. Vis. Image Underst.* 119, 15–26.
- Swendsen, R.H., Wang, J.-S., 1987. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.* 58 (2), 86–88.
- Tu, Z., Zhu, S.-C., 2002. Image segmentation by data-driven Markov chain Monte Carlo, pattern analysis and machine intelligence. *IEEE Trans.* 24 (5), 657–673.
- Velasco, F., 1980. Thresholding using the ISODATA clustering algorithm. *IEEE Trans. Syst. Man Cyb.* 10, 771–774.
- Vese, L., Chan, T., 2002. A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vision* 50, 271–293.
- Watanabe, H., Sashida, S., Okabe, Y., Lee, H., 2011. Monte Carlo methods for optimizing the piecewise constant Mumford-Shah segmentation model. *New J. Phys.* 13, 023004.
- Zeng, X., Chen, W., Peng, Q., 2006. Efficiently Solving the Piecewise Constant Mumford-Shah Model Using Graph Cuts. Zejiang University, P.R. China Technical report, Department of Computer Science.
- Zhai, Y., Shah, M., 2006. Video scene segmentation using Markov chain Monte Carlo, multimedia. *IEEE Trans.* 8 (4), 686–697.