

Content-Based Coding of Volume Data Using Finite Mixture Model, Octree and Shape-Adaptive DCT

N. ICHIMURA

A content-based coding method of volume data is proposed in this paper. The proposed method is based on “texture-contour” approach. First, volume data is segmented, then shape, texture and opacity information of contents are coded. Each processing is carried out using methods and the representations as follows: (i)segmentation: probability density function estimation using finite mixture model, (ii)shape: depth-first (DF) representation of octree, (iii)texture: shape-adaptive DCT (SA-DCT) and entropy coding, (iv)the parameters of finite mixture model. Using content-based coding, we can obtain shape of object which is useful in visualization of decoded data. Experimental results for CT data are shown.

§ 1 Introduction

Volume data is sampled function of three spatial dimensions. It is useful to represent internal structure of object; a typical example is data generated from a set of CT images. However, the size of volume data is normally very large. Thus coding method for volume data is needed. Since volume data has three dimensional data structure, visualization method is needed to display contents in it. To visualize desired contents properly, contents should be segmented. Thus segmentation is also needed.

Coding method for volume data has been developed based on waveform coding using vector quantization¹⁾, DCT²⁾, wavelet³⁾, fractal⁴⁾, etc. However, not only waveform coding but also *content-based coding* is suitable for volume data because segmentation is indispensable in visualization of volume data (**Fig.1**).

A region-based coding method of volume data is proposed in this paper. An overview of the proposed method is shown in **Fig.2**. The proposed method is based on texture-contour approach⁵⁾; volume data is segmented, then shape, texture and opacity information of contents are coded.

Volume data is segmented into 3D regions via estimation of probability density function of the data using finite mixture model. The contents to be coded are ex-

tracted by selecting the 3D regions. The shape of the contents is represented by octree¹²⁾ and coded using depth first(DF) representation¹³⁾ of it. Octree divides the shape of the contents into blocks. The texture in each block is transformed by shape-adaptive DCT(SA-DCT) and coded using transform coefficients.

Volume rendering⁸⁾ is used in visualization of decoded data. Opacity of each region used in volume rendering is represented as a posteriori probability calculated from the parameters of finite mixture model used in segmentation.

An advantage of the proposed algorithm is combination of octree and SA-DCT. To construct octree, contents are divided into the blocks with the size of power of 2. All locations of the blocks are hierarchically stored in octree. Thus the locations of the blocks

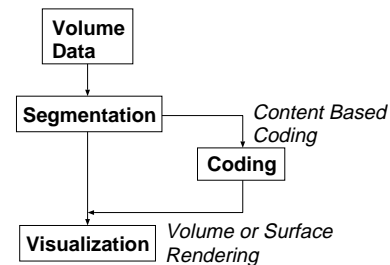


Fig. 1 Relationship among segmentation, visualization and coding of volume data.

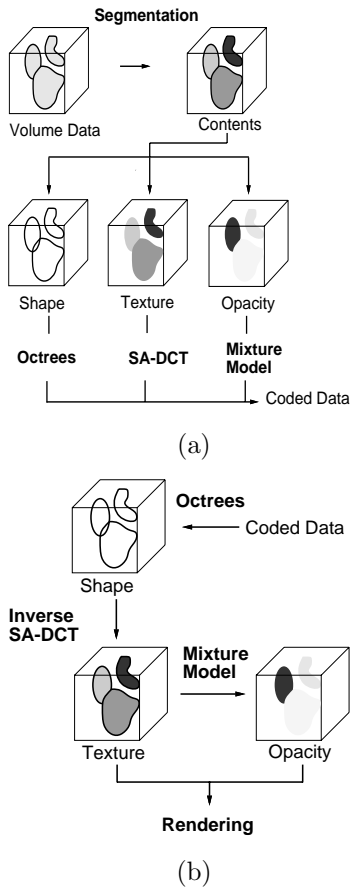


Fig. 2 Outline of the proposed coding method. (a)Encoding. (b)Decoding.

to be transformed by shape-adaptive manner can be easily searched using octree; the blocks on region boundary can be detected as gray nodes in octree.

Section 2 explains volume rendering used in visualization of decoded data, and the relationship between segmentation and volume rendering. Section 3 presents the segmentation method using finite mixture model. Section 4 describes the coding of shape using DF representation of octree. Section 5 shows the coding of texture by SA-DCT. Section 6 shows the coding of opacity using the parameters of finite mixture model. Section 7 demonstrates the experimental results for CT data. Section 8 presents conclusions.

§ 2 Volume Rendering

In volume rendering, the portions to be visualized in volume data should be segmented to be assigned high opacity. After segmentation, the colors of voxels calculated by shading process are projected onto image plane $I(X, Y)$ through transparencies added from back to front along viewing ray (Fig.3). This process can

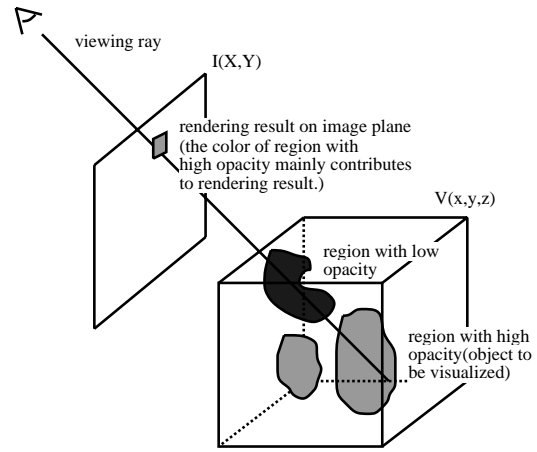


Fig. 3 Volume rendering.

be represented as follows:

$$I(X, Y) = \sum_{i=1}^K \left[c_i o_i \prod_{j=i+1}^K (1 - o_j) \right] \quad (1)$$

where c_i and o_i are color and opacity of the i -th voxel ($i = 1, \dots, K$) which intersects viewing ray. Obviously, segmentation is indispensable to visualize desired portions in volume data by volume rendering.

§ 3 Region Segmentation Using Finite Mixture Model^{6),7)}

Volume data is segmented by a feature space approach. The distribution of a set of the feature vectors $V = \{\mathbf{x}_i\}_{i=1}^N$ obtained from volume data is represented by the following finite mixture model.

$$f(\mathbf{x} | \Theta) = \sum_{j=1}^r \lambda_j \alpha_j(\mathbf{x} | \theta_j) \quad (2)$$

where \mathbf{x} is a feature vector in \mathbf{R}^n , r is the number of component densities $\alpha_j(\cdot)$, $\{\lambda_j\}_{j=1}^r$ are mixing proportions such that $\sum_{j=1}^r \lambda_j = 1$, $0 \leq \lambda_j \leq 1$, θ_j is a set of parameters of $\alpha_j(\cdot)$, and $\Theta = \{\{\lambda_j\}_{j=1}^r, \{\theta_j\}_{j=1}^r\}$ is a set of the parameters.

A multivariate t distribution⁹⁾ is used as a component.

$$\alpha(\mathbf{x} | \theta) = |\mathbf{V}^{-1}| h(d_m^2 | \nu) \quad (3)$$

$$h(d_m^2 | \nu) = \frac{\Gamma\{(\nu + n)/2\}}{\{\Gamma(1/2)\}^n \Gamma(\nu/2) \nu^{n/2}} \cdot (1 + d_m^2/\nu)^{-(\nu+n)/2} \quad (4)$$

$$d_m^2 = (\mathbf{x} - \mathbf{m})^t \mathbf{V}^{-1} (\mathbf{x} - \mathbf{m}) \quad (5)$$

where $\theta = \{\mathbf{m}, \mathbf{V}, \nu\}$, and \mathbf{m} ($n \times 1$), \mathbf{V} ($n \times n$) and ν are location, scatter matrix and degree of free-

dom, respectively. This distribution is identical to the normal distribution when $\nu \rightarrow \infty$, and can represent more heavy tailed distribution by changing ν .

The parameters of the finite mixture model are estimated by a maximum likelihood method. The evaluation function for parameter estimation is as follows:

$$E = \sum_{i=1}^N \log \sum_{j=1}^r \lambda_j \alpha_j(\mathbf{x}_i | \theta_j) - \xi \left(\sum_{j=1}^r \lambda_j - 1 \right) \quad (6)$$

where ξ is a Lagrange multiplier. Expectation-Maximization(EM) algorithm^{10),11)} can be used to maximize above evaluation function. The EM algorithm for the current model is as follows:

[Step 1(E-step)] Calculate a posteriori probability $P(S_j | \mathbf{x}_i)$ for the j -th component S_j and a weight $w(d_{mi}^2 | \nu^c)$ under the current approximation of the parameters $\Theta^c = \{ \{ \lambda_j^c \}_{j=1}^r, \{ \theta_j^c \}_{j=1}^r \}$.

$$P(S_j | \mathbf{x}_i) = \frac{\lambda_j^c \alpha_j(\mathbf{x}_i | \theta_j^c)}{f(\mathbf{x}_i | \Theta^c)} \quad (7)$$

$$w(d_{mi}^2 | \nu_j^c) = -2 \frac{\partial \log h(d_{mi}^2 | \nu_j^c)}{\partial d_{mi}^2} = \frac{\nu + n}{\nu + d_{mi}^2} \quad (8)$$

$$d_{mi}^2 = (\mathbf{x}_i - \mathbf{m}_j^c)^t \mathbf{V}_j^{c-1} (\mathbf{x}_i - \mathbf{m}_j^c) \quad (9)$$

$i = 1, \dots, N; j = 1, \dots, r$

[Step 2(M-step)] Calculate the parameters Θ^+ .

$$\lambda_j^+ = \frac{1}{N} \sum_{i=1}^N P(S_j | \mathbf{x}_i) \quad (10)$$

$$\mathbf{m}_j^+ = \frac{\sum_{i=1}^N P(S_j | \mathbf{x}_i) w(d_{mi}^2 | \nu_j^c) \mathbf{x}_i}{\sum_{i=1}^N P(S_j | \mathbf{x}_i) w(d_{mi}^2 | \nu_j^c)} \quad (11)$$

$$\mathbf{V}_j^+ = \frac{\sum_{i=1}^N P(S_j | \mathbf{x}_i) w(d_{mi}^2 | \nu_j^c) (\mathbf{x}_i - \mathbf{m}_j^+) (\mathbf{x}_i - \mathbf{m}_j^+)^t}{\sum_{i=1}^N P(S_j | \mathbf{x}_i)} \quad (12)$$

$$j = 1, \dots, r$$

[Step 3(M-step)] Calculate the parameters $\{ \nu_j^+ \}_{j=1}^r$ maximizing eq.(6) using the quasi-Newton method under $\lambda_j^+, \mathbf{m}_j^+, \mathbf{V}_j^+$ calculated in Step 2.

[Step 4] If

$$|l(\Theta^c) - l(\Theta^+)| < \epsilon \quad (13)$$

is satisfied or the number of iteration exceeds a given number, then terminate this algorithm. Otherwise, update Θ^c by Θ^+ and go to Step 1.

The maximum likelihood solution for ν can not be

represented in closed-form, so the numerical search in Step 3 is needed.

The weight of eq.(8) shows that the data far from the location of the t distribution have small effect for parameter estimation. That is, segmentation using this finite mixture model can reduce the effect of outliers in parameter estimation.

Since the component density is corresponded to region in volume data, volume data is segmented via this probability density function estimation; a posteriori probability given by eq.(7) is used to segment volume data. The component densities corresponding to the regions to be coded are selected, and a posteriori probabilities of *all* voxels are calculated using them. Then contents are segmented by extracting the voxels with larger a posteriori probability than a given threshold.

§ 4 Shape Coding by Octree

Encoding, decoding and processing for only desired portions are important requirements for content-based coding. Thus the representations which can be used to search and compress the locations of contents are desirable to represent shape. Octree¹²⁾ is one of such representations.

Octree is a solid model which is constructed by recursive subdivision of 3D regions into the entities composed by eight cubes (**Fig.4(a)**). 3D regions divided into blocks with the size of power of 2; major portions are represented by the blocks with large size, e.g., $32 \times 32 \times 32$, while detailed portions are represented by the blocks with small size, e.g., $4 \times 4 \times 4$. The structure of octree is represented by a tree including nodes with eight children (**Fig.4(b)**). All the locations of contents are hierarchically stored in the tree; search of the locations of contents can be easily carried out as one of the tree.

The nodes of octree have three types: black, white and gray. The blocks corresponding to black and white nodes have “interesting” and “uninteresting” portions. The blocks corresponding to gray nodes include both “interesting” and “uninteresting” portions; gray node represents boundary of contents.

Octree is constructed from the binary volume data obtained as segmentation result. Samet’s algorithm¹⁴⁾ for quadtree is extended to construct octree. DF representation¹³⁾ is used as a compressed form of

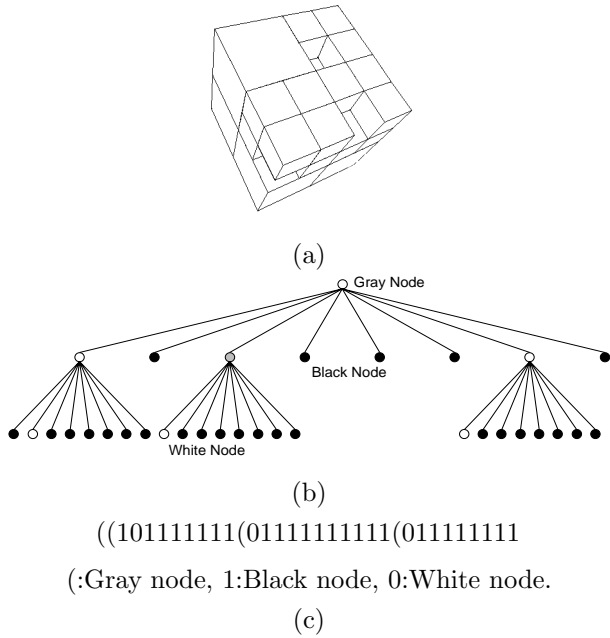


Fig. 4 An example of octree. (a)Object. (b)Octree of the object. (c)DF representation of the octree.

octree. In DF representation, the node types, i.e. black, white and gray, are ordered by depth first manner(**Fig.4(c)**). Shape information is coded by assigning binary code to each symbol in DF representation.

§ 5 Texture coding by SA-DCT

Texture in the blocks of octree is coded via orthogonal transformation and entropy coding of transform coefficients. DCT is used for the blocks in contents, while SA-DCT is used for the blocks on the boundary of contents. The locations of the blocks can be searched using octree. By this search, the 3D regions are divided into the blocks with the size corresponding to the searched level of octree.

A SA-DCT used in this paper is the extension of 2D version¹⁵⁾. A variable length DCT basis is used in the SA-DCT. The length of the basis is the same as one of the region on each axis. The basis with the length M, DCT-M, is given as follows:

$$\text{DCT-M}(u, k) = a_0 \cdot \cos \left[\frac{(2k + 1) u \pi}{2M} \right] \quad (14)$$

$$a_0 = \begin{cases} 1/\sqrt{2}, & u = 0 \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

$k, u = 0, \dots, M - 1$

where u and k are spatial frequency and coordinate of voxel, respectively. The DCT coefficient $c_{x,y,z}$ for

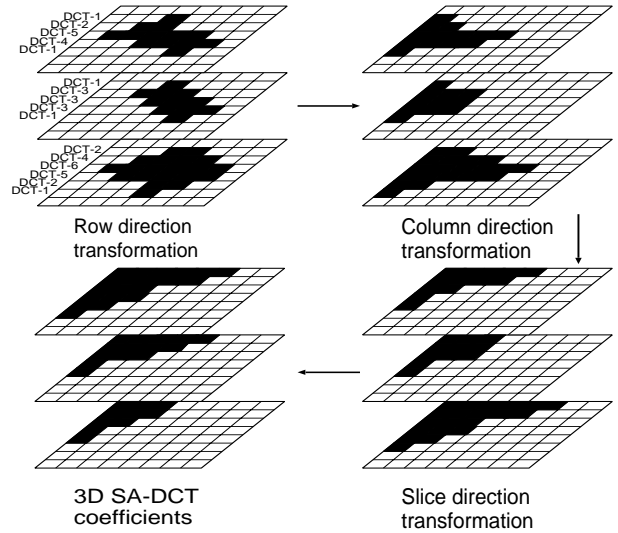


Fig. 5 Shape-Adaptive DCT.

local coordinate (x, y, z) in the block is calculated using successive 1D transformation for each axis as shown in **Fig.5**.

The coefficients are quantized as follows:

$$c_{x,y,z}^q = \text{round} \left(\frac{c_{x,y,z}}{s \cdot q_{x,y,z}} \right) \quad (16)$$

where s is scale factor, $q_{x,y,z}$ is quantization step size and round() is round-off operation.

The quantized coefficients are coded via Zig-Zag scanning, run-length coding and entropy coding like JPEG.

§ 6 Opacity Coding by Finite Mixture Model

The opacity of each voxel used in volume rendering is represented by the parameters of the finite mixture model. A posteriori probability calculated by eq.(7) shows the membership of each voxel for contents. Thus a posteriori probability can be used as the opacity. In decoding process, the opacity is calculated using reconstructed data and the parameters of the finite mixture model(**Fig.2(b)**).

§ 7 Experimental Results

The CT data of a head(128×128×128, 12[bit/voxel]) and a wrist(128×128×128, 8[bit/voxel]) were used. The block size of the SA-DCT was 8×8×8. The quantization step for the DCT coefficients was determined by the sum of *local* coordinate values $v = x + y + z$. The quantization table is shown in **Table 1**.

Table 1 Quantization table for DCT coefficients.

v	0	1	2	3	4	5	6
$q_{x,y,z}$	16	12	12	14	17	26	34
v	7	8	9	10	11	12	>12
$q_{x,y,z}$	55	64	80	85	90	100	100

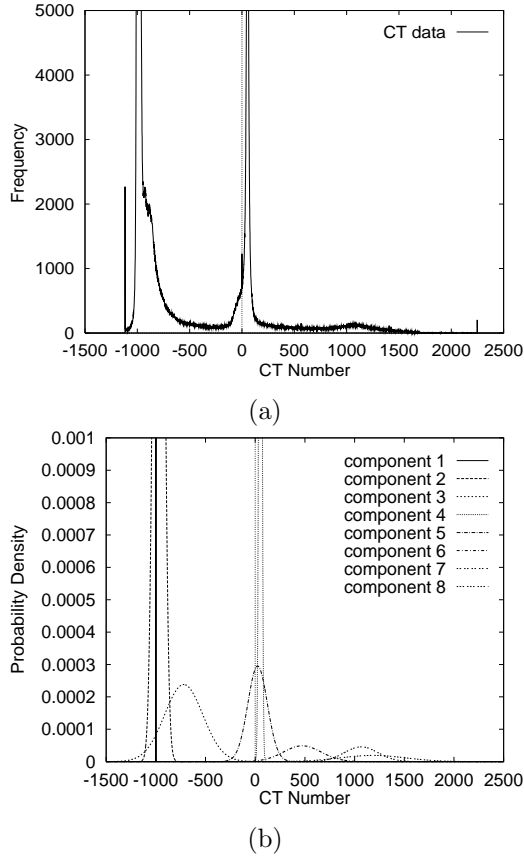


Fig. 6 Estimation result of the probability density function for the histogram of the head data by the finite mixture model with 8 components. (a) histogram of the CT number. (b) estimated probability density function using the finite mixture model.

7.1 Segmentation Results

The feature used in segmentation was the CT number of each voxel (**Fig. 6(a)** and **Fig. 7(a)**). The probability density function of the CT number was estimated by the finite mixture model (**Fig. 6(b)** and **Fig. 7(b)**). The 8 and 10 components were used for the head and the wrist data, respectively; these components were needed to segment the volume data into meaningful portions.

In the head data, the components covered the range $[-1500, 0]$, $[-400, 500]$ and $[0, 2000]$ in the histogram of the CT number mainly corresponded to air, soft tis-

sue and skull, respectively (**Fig. 6(b)**). The contents to be coded were the soft tissue and the skull. Hence, a posteriori probabilities of all voxels were calculated using the components covered the range $[-400, 2000]$. Then voxels with larger probability than the threshold 0.1 were extracted. In this case, the contents can be segmented by simple thresholding. However, the probability density function of the CT number can not be obtained; the opacity used in visualization can not be calculated.

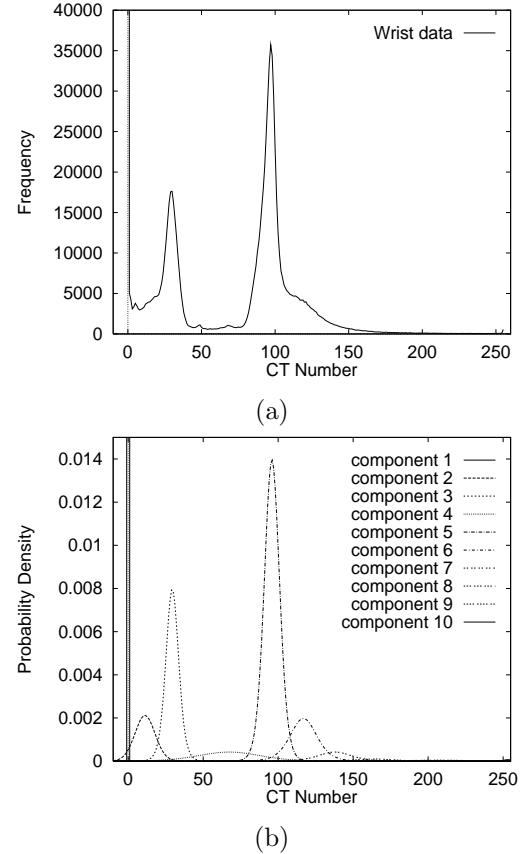


Fig. 7 Estimation result of the probability density function for the histogram of the wrist data by the finite mixture model with 10 components. (a) histogram of the CT number. (b) estimated probability density function using the finite mixture model.

In the wrist data, the components covered the range $[0, 30]$, $[20, 120]$ and $[190, 250]$ in the histogram mainly corresponded to air, skin and muscle and bone, respectively (**Fig. 7(b)**). The components covered the range $[20, 250]$ were used to calculate the probability. The segmentation was carried out the same manner as the head data.

7.2 Coding Results

The shape of the contents of the head data was represented by the octree(**Fig.8**). The slice of the decoded data shows that the shape and the texture were properly coded(**Fig.9**). To render the decoded data, the opacities were calculated by eq.(7) using the parameters of the finite mixture model included in the coded data. The rendering results of the decoded data showed that the major features of the contents were sufficiently reconstructed(**Fig.10**). The approximation error of the texture leads to the semi-transparent region appeared in the rendering results; the opacities assigned to the voxels are varied with the CT numbers of the decoded data with the error due to the quantization of DCT coefficients. Furthermore, the discontinuity between “interesting” and “uninteresting” portions lead to an artifact; the smoothness of the surface in the rendering results were lost due to the discontinuity(**Fig.10**)

The SA-DCT was useful to improve a coding performance; the compression ratio and the PSNR of the SA-DCT were superior to that of the DCT(**Table 2**).

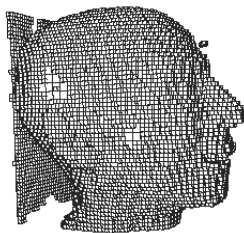


Fig. 8 The octree of the CT data of the head.

Table 2 Coding performance for the CT data of the head.

Method	Scale factor	PSNR [dB]	C.R.1	C.R.2
SA-DCT	1	35.9	7.3	32.3
	3	32.0	10.4	46.0
	5	30.6	11.8	52.2
DCT	1	35.0	5.8	25.7
	3	31.0	8.1	35.9
	5	29.5	9.1	40.3

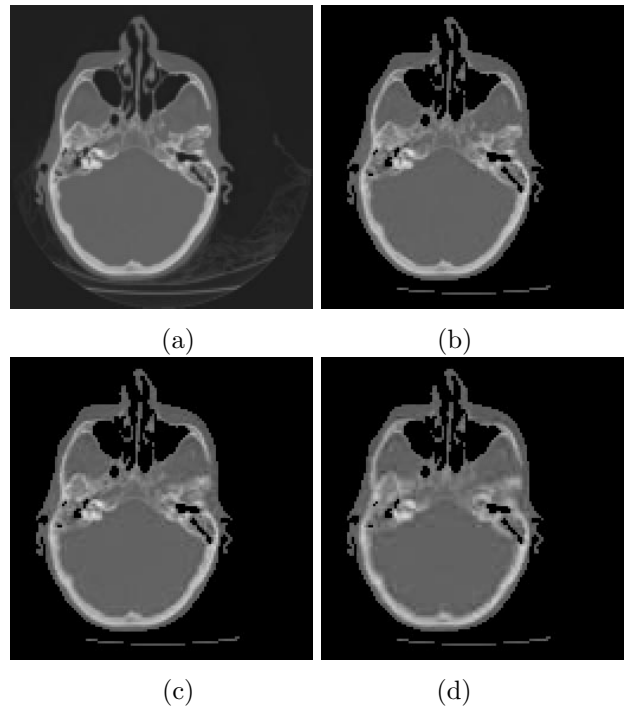


Fig. 9 Experimental result for the CT data of the head. The comparison on the slice image. (a)Original slice. (b)Scale factor:0.5. (c)Scale factor:1. (d)Scale factor:3.

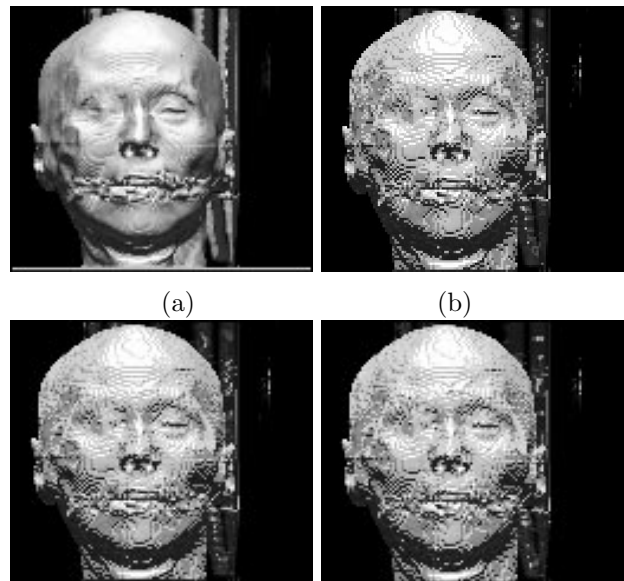
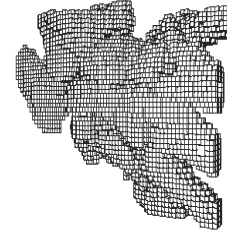


Fig. 10 Experimental result for the CT data of the head. The comparison on the rendering result. (a)Original data. (b)Scale factor:0.5. (c)Scale factor:1. (d)Scale factor:3.

Table 3 Required bits for each information of the CT data of the head.

Method	Scale factor	Shape	Texture	Opacity
SA-DCT	1	158880	618823	1037
	3		386829	
	5		321873	
DCT	1	158880	818642	1037
	3		541732	
	5		463965	

**Fig. 11** The octree of the CT data of the wrist.**Table 4** Coding performance for the CT data of the wrist.

Method	Scale factor	PSNR [dB]	C.R.1	C.R.2
SA-DCT	0.1	35.6	2.0	61.1
	0.4	31.2	2.7	84.4
	0.8	29.1	3.1	95.1
DCT	0.1	34.2	1.4	42.6
	0.4	29.0	1.8	56.5
	0.8	26.5	2.0	62.3

Table 5 Required bits for each information of the CT data of the wrist.

Method	Scale factor	Shape	Texture	Opacity
SA-DCT	0.1	124324	149124	1295
	0.4		73242	
	0.8		50863	
DCT	0.1	124324	268294	1295
	0.4		171112	
	0.8		143481	

C.R.1 and C.R.2 in **Table 2** are defined as follows:

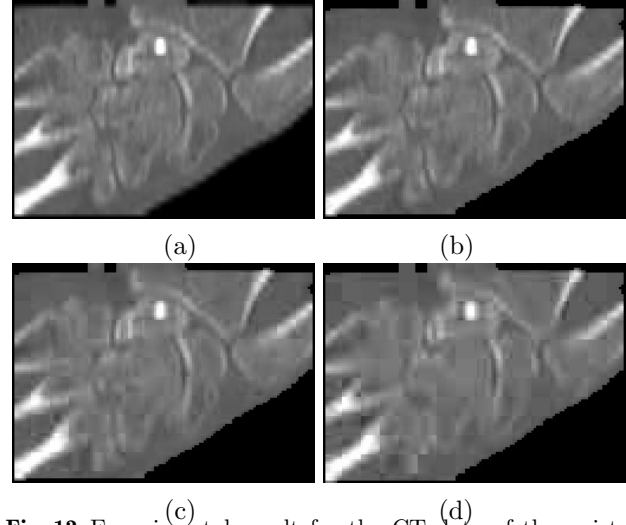
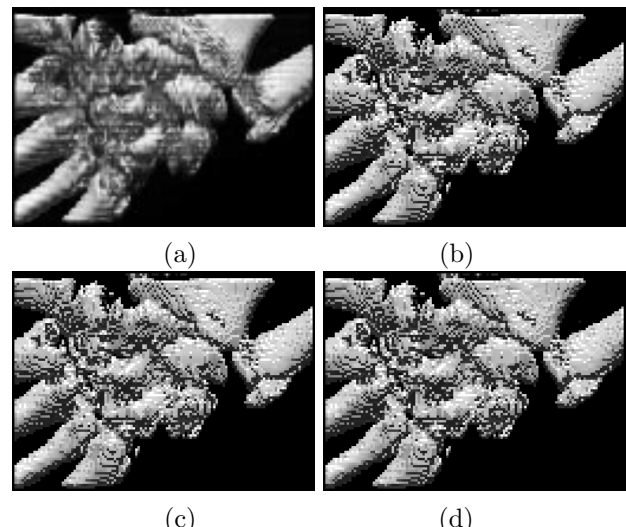
$$C.R.1 = \frac{\text{number of bits in coded contents}}{\text{number of bits in contents}} \quad (17)$$

$$C.R.2 = \frac{\text{number of bits in coded contents}}{\text{number of bits in volume data}} \quad (18)$$

The data reduction by segmentation is taken into account in C.R.2.

The amount of bits for the texture was larger than one for the shape and the opacity(**Table 3**). Thus the use of the SA-DCT was important to improve the coding performance.

Almost the same results were obtained for the CT data of the wrist (**Fig.11–Fig.13**, **Table 4** and **Table 5**).

**Fig. 12** Experimental result for the CT data of the wrist. The comparison on the slice image. (a)Original slice. (b)Scale factor:0.1. (c)Scale factor:0.4. (d)Scale factor:0.8.**Fig. 13** Experimental result for the CT data of the wrist. The comparison on the rendering result. (a)Original data. (b)Scale factor:0.1. (c)Scale factor:0.4. (d)Scale factor:0.8.

§ 8 Conclusions and Future Works

The content-based coding of volume data has been proposed. The feature of the proposed algorithm is the combination of octree and coding of the texture using SA-DCT; the search of the locations of the blocks transformed by shape-adaptive manner can be easily carried out as one of octree. The experimental results for the CT data showed the usefulness of the proposed method.

In the experiments, bits were allocated only “interesting” portions. However if we allocates bits for “uninteresting” portions, the boundary between the two portions becomes smooth; the quality of the visualization result of decoded data can be improved. Moreover we may allocate bits *adaptively* for the portions by switching the coding parameters, i.e., scaling factor, to preserve compression ratio while improving the quality of visualization. The preliminary results of the “adaptive bits allocation” are shown in **Fig.14** and **Fig.15**. The smoothness of the surface in the rendering results was improved. However, the blocking distortions in “uninteresting” portions were appeared. More detailed consideration for adaptive bits allocation remains as an important future work.

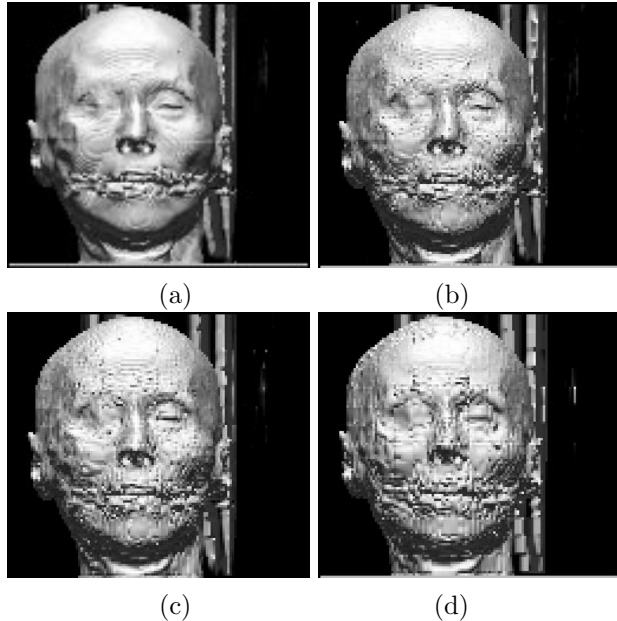


Fig. 14 Experimental result of the adaptive bit allocation for the CT data of the head. (a)original. (b)Scale factor: inside:0.5, outside:1. (c)Scale factor: inside:1, outside:2. (d)Scale factor: inside:3, outside:4.

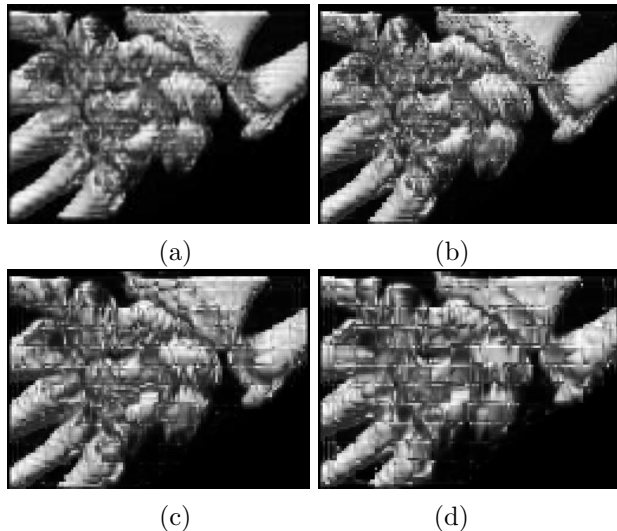


Fig. 15 Experimental result of the adaptive bit allocation for the CT data of the wrist. (a)Original. (b)Scale factor: inside:0.1, outside:0.2. (c)Scale factor: inside:0.4, outside:0.6. (d)Scale factor: inside:0.8, outside:1.0

Acknowledgments

The author would like to thank Dr. Nobuyuki Otsu, Director of the Machine Understanding Division, for his encouragement regarding this research. Thanks also to Dr. Shigeru Muraki for useful discussions at the beginning of this research.

References

- P. Ning and L. Hesselink(1993) “Fast Volume Rendering of Compressed Data,” Proc. of IEEE Visualization '93, pp.11-18.
- B. Yeo and B. Liu(1995) “Volume Rendering of DCT-Based Compressed 3D Scalar Data,” IEEE Trans. on Visualization and Computer Graphics, **Vol.1**, No.1, pp.29-43.
- S. Muraki(1993) “Volume data and wavelet transforms,” IEEE Computer Graphics & Applications, **Vol.13**, No.4, pp.50-56.

- W.O. Cochran, J.C. Hart and P.J. Flynn(1996)
 “Fractal Volume Compression,” IEEE Trans. on Visualization and Computer Graphics, **Vol.2**, No.4, pp.313-322.
- M.Kunt,A.Ikonomopoulos and M.Kocher(1985)
 “Second-Generation Image-Coding Techniques,” Proc. IEEE, **Vol.73**, No.4, pp.549-574.
- N. Ichimura(1996) “Volume Data Coding Based on Region Segmentation Using Finite Mixture Model, ” Proc. IEEE International Conference on Image Processing, **Vol.III**, pp.363-366.
- N. Ichimura(1996) “Region Based Coding of Volume Data Using Finite Mixture Model,” Proc. 7th British Machine Vision Conference, pp.183-192.
- M. Levoy(1988) “Display of Surfaces from Volume Data,” IEEE CG & A, **Vol.8**, No.5, pp.29-37.
- K.L. Lange, R.J.A. Little and J.M.G. Taylor(1989)
 “Robust Statistical Modeling Using the t Distribution,” Journal of the American Statistical Association, **Vol.84**, No.408, pp.881-896.
- A.P. Dempster, N.M. Laird and D.B. Rubin(1977)
 “Maximum Likelihood from Incomplete Data via the EM Algorithm,” J.R.Statist.Soc. B, **Vol.39**, No.1, pp.1-38.
- R.A. Redner and H.F. Walker(1984) “Mixture Densities,Maximum Likelihood and the EM Algorithm,” SIAM Revis, **Vol.26**, No.2, pp.195-239.
- H.H.Chen and T.S.Huang(1988) “A Survey of Construction and Manipulation of Octrees,” CVGIP, **Vol.43**, pp.409-431.
- E.Kawaguchi and T.Endo(1980) “On a Method of Binary-Picture Representation and Its Application to Data Compression,” IEEE Trans. PAMI, **Vol.2**, No.1, pp.27-35.
- H.Samet(1980) “Region Representation: Quadrees from Binary Arrays,” CGIP, **Vol.13**, pp.88-93.
- T.Sikora and B. Makai(1995) “Shape-Adaptive DCT for Generic Coding of Video,” IEEE Trans. Circuits and Systems for Video Technology, **Vol.5**, No.1, pp.59-62.

(Accepted December 24, 1998)

The Author



Naoyuki ICHIMURA

Bottom-up Modeling Lab., Machine Understanding Division

E-mail: ichimura@etl.go.jp

His current research interests are computer vision and data compression.