Vol. 38 No. 8 Transactions of Information Processing Society of Japan Aug. 1997

Regular Paper

A JPEG Codec Adaptive to the Relative Importance of Regions in an Image

JIYING ZHAO,† YOSHIHISA SHIMAZU,†† KOJI OHTA,†††
RINA HAYASAKA† and YUTAKA MATSUSHITA†

This paper presents a novel standard-baseline-JPEG-compatible method for still image compression that allows flexible control of the decompression quality of regions in an image. In this way, it not only represents the important parts of the image with higher quality, but also reduces the bit rate as a whole and reflects the human sense of what is important in an image. We use fuzzy technique to determine important regions in a human-like manner, allocating a higher bit rate to important regions and a lower bit rate to the other parts. This paper introduces and describes methods for determining important regions, representing and filling arbitrarily shaped regions, and compressing and decompressing images while adapting to the relative importance of the regions. It also describes some experimental results obtained by using the proposed method.

1. Introduction

The goal of image compression techniques is to reduce redundancy and minimize the bit count of digitized images while maintaining an acceptable quality that crucially depends on the human visual system (HVS).

Recently, JPEG coding systems have been widely used in diverse applications in still image compression. In baseline JPEG syntax[1], first, each component of an image is grouped into 8×8 pixel blocks regardless of the image content. Each block is then independently transformed by an 8×8 forward-discrete-cosine-transform (FDCT), and quantized with an 8×8 matrix of uniform scalar quantizer step sizes, known as the Q-matrix. The quantized blocks are subsequently entropy-coded in raster scan order. The decoding is the inverse of the encoding, and consists of entropy decoding, dequantization, and inverse DCT[1)~3)]. Applications must make a trade-off between image quality and compression ratio. A higher compression ratio may mean lower decoded image quality, while higher decoded image quality may mean a lower compression ratio.

Some block-transform-based studies have addressed scene-adaptive or image-adaptive issues[3),4)], but have concentrated on either stabilizing the bit rate by adapting to the scene of the image, or optimizing the whole decoding quality by adapting to images. In other words, they have not considered the importance of regions in images. Recently, there has been some renewed interest in segmentation-based image coding[5),6)]. In these methods, an image is first segmented into a number of regions, and the region texture and boundary information is then coded. Segmentation-based techniques are considered very promising, and have received considerable attention in MPEG4 standardization[7)], but the strong popularity of JPEG coding systems must be taken into consideration. Our research goal is to make JPEG adaptive to the relative importance of regions in an image.

Looking at a picture, humans generally only pay attention to important objects, while ignoring less important ones in the same picture. Normally, for a camera, there is a focal distance and depth, and objects in the photograph that are focused on or within the depth will be clear, while others will not. In a meaningful picture, there is normally a subject that is focused on. To take another example, in a video conference, only the speaker on whom the camera is focused in an image is important, while whatever is behind him, whether it is a map or shelf book or whatever, is not important. These considerations motivated us to develop a new compression method, an importance-adaptive JPEG codec, which uses fuzzy technique to segment out important regions in the same way as humans do, and compresses them with higher quality, while compressing unimportant regions and the background with lower quality. The method is attractive because it not only repre-

† Department of Instrumentation Engineering, Faculty of Science and Technology, Keio University
†† Matsushita Electric Industrial Co. Ltd.
††† Sony Corporation

sents important parts with higher quality, but also reduces the bit rate as a whole and reflects the human sense of what is important in an image.

The following sections will introduce the proposed importance-adaptive JPEG codec scheme and its implementation.

2. Importance-adaptive Codec Scheme

The new scheme is designed in accordance with the JPEG standard shown in **Fig. 1**.

Let us briefly review the quantization procedure in JPEG [1),2)]. After output from the FDCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a 64-element quantization table, which must be specified by the application (or user) as an input to the encoder. Each element can be any integer from 1 to 255, which specifies the step size of the quantizer for its corresponding DCT coefficient. The purpose of quantization is to discard information that is not visually significant. Quantization is a many-to-one mapping, and therefore is fundamentally lossy. It is the principle source of lossiness in DCT-cased encoders. The 64 integer elements of the JPEG Q-matrix largely determine the quality and compression of JPEG-coded images [4)].

Quantization is defined as division of each DCT coefficient by its corresponding quantizer step size, followed by rounding to the nearest integer:

$$F^{Q}(u,v) = \mathit{Integer\ Round}\left(\frac{F(u,v)}{Q(u,v)}\right). \tag{1}$$

This output value is normalized by the quantizer step size. Dequantization is the inverse function, which means simply that the normalization is removed by multiplying by the step size, which returns the result to a representation appropriate for input to the IDCT:

$$F^{Q'}(u,v) = F^{Q}(u,v) * Q(u,v). \tag{2}$$

In our scheme, the "example" quantization tables K.1 and K.2 of the JPEG standard [1)] are used, with the scale-size *Scale* trading off compressed file size against the quality of the reconstructed image.

$$Q(u,v) = Q^{E}(u,v) * \frac{\mathit{Scale}}{50}, \tag{3}$$

where Q is the expected quantization table, Q^E is the "example" quantization table K, and *Scale* is the numerator used to scale the Q^E.

From Eq. (3), it is clear that the compression ratio could be controlled by changing the *Scale*. The larger the *Scale*, the larger the step size, the higher the compression ratio, and hence the lower the quality. *Scale* = 50 means that just the Q^E is used.

The proposed scheme segments an image into regions, determines the importance of each region, forms an MDU-Map to represent the importance of each minimum data unit (MDU) in the image, then accordingly compresses or decompresses the image by a method that will be introduced in the following sections.

3. Clustering-based Segmentation

The proposed coding scheme is adaptive to the relative importance of regions, and therefore it is necessary to determine the importance of each region. For this, image segmentation is indispensable. Despite the wide variety of image segmentation techniques, there are still some problems, for example, the user must select the method to be used as well as the input, and adjust the parameters as needed [8),9)]. Our segmentation scheme [10)] is based on a faster, wholly automatic, non-parametric clustering algorithm [11)], so no interaction with the user is needed. Our main objective is to produce a segmentation result as near as possible to human judgment, in order to improve the importance determination and codec efficiency, and reduce the bit count for region representation. For this reason, we first carry out over-segmentation based on clustering, and then form a reasonable region set through merging based on fuzzy reasoning and other techniques, which reflect human *a priori* knowledge of images.

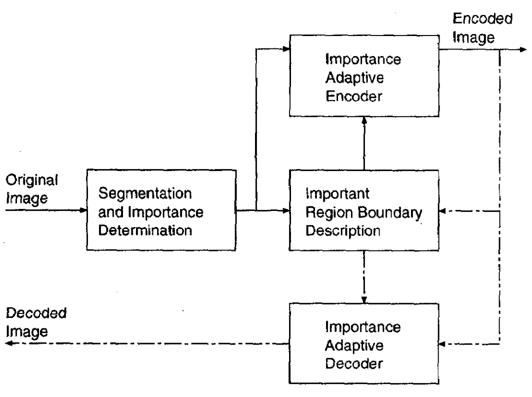


Fig. 1 Importance-adaptive image encoding and decoding scheme.

3.1 Preprocessing

As a preprocess, the original image is transformed from RGB into CIE $L^*a^*b^*$ representation, and then all $L^*a^*b^*$ components are contracted. After contraction, 3×3 median-filtering is used, in which an input pixel is replaced by the median of that pixel and its 8 neighbors. The purpose of the preprocessing is to eliminate noise and to average the texture of the original image, and also to reduce the computational load.

3.2 Peak-climbing-algorithm-based Clustering

The algorithm we used is a non-parametric-histogram-based clustering algorithm [11], whose principle is to locate local modes (peaks) in the histogram made by multi-dimensional feature vectors.

The first step in generating the histogram is to quantize the cubic color space into non-overlapping cells. As a result, each cell will encompass a certain number of data points. The histogram is generated by counting the number of data points falling into each cell.

By searching for the largest count of the neighbors of a particular count, a link is established between the particular count and the largest count when the largest is larger than the particular. If two or more neighboring cells have the same count for a particular cell, one of them is arbitrarily chosen as a parent. In other words, a particular count chooses the largest count in the neighborhood as its parent. A peak is defined as a count that has no parent. Each peak and the counts linked to it are taken as a distinct cluster.

We set the cluster number C to 20, so as not to make segmentation too busy, and assign an appropriate initial value for the side length D of a cube.

- Divide the space into cubes (cells) with side length D, create a histogram, and do peak-climbing on the CIE $L^*a^*b^*$ space,
- Decrease the size D by a certain value and loop back to the former step; repeat this step until the cluster number C is reached.

When the cluster number reaches C, the clustering is finished, clusters are formed, and regions are extracted.

3.3 Region Merging and Labeling

Write the clusters in the CIE $L^*a^*b^*$ space back into the original image. After smoothing using the mode filter, regions whose pixel counts are less than half of what they were before smoothing are considered as meaningless regions, and are combined with their neighbors. The mode filtering used here is a procedure that replaces one pixel with the most common value of that pixel and its 8 neighbors.

After mode filtering, the labeling is carried out on the spatial space, not the color space, to ensure that each region has a unique number. Then, the labeling value is mode-filtered to further merge meaningless clusters.

3.4 Fuzzy-reasoning-based Region Merging

Regions with the same color can be taken as a single region [14]. Taking account of contiguous relationships and color differences between connected regions, fuzzy reasoning is used to further merge meaningless regions. The following three parameters are the color difference between $region_m$ and its adjacent $region_n$ ($m \neq n$), and are used as the inputs of the fuzzy reasoning:

- **$L^*a^*b^*$ color difference of a whole region.** This is defined as follows:

$$\begin{aligned}\Delta_{color} &= ||color_m - color_n|| \\ &= \sqrt{(\Delta_L)^2 + (\Delta_a)^2 + (\Delta_b)^2},\end{aligned} \tag{4}$$

  where $\Delta_L = \overline{L^*_m} - \overline{L^*_n}$, $\Delta_a = \overline{a^*_m} - \overline{a^*_n}$, $\Delta_b = \overline{b^*_m} - \overline{b^*_n}$. Here, $(\overline{L^*}, \overline{a^*}, \overline{b^*})$ is the representative color of the region, and $\overline{L^*}, \overline{a^*}, \overline{b^*}$ are respectively defined as the mean values of the L^*, a^*, b^* values for the whole region.
- **Hue angle difference of a whole region.** Since light components are easily affected by shadows and shading, only the hue angle differences are used; that is, $\Delta H^\circ_{ab} = \overline{H^\circ_{abm}} - \overline{H^\circ_{abn}}$. Here, $\overline{H^\circ_{ab}}$ is defined as follows:

$$\overline{H^\circ_{ab}} = \frac{180^\circ}{\pi} \tan^{-1}\left(\frac{\overline{b^*}}{\overline{a^*}}\right),$$

  where $\overline{a^*}$, $\overline{b^*}$ are respectively the mean values of the a^*, b^* values of the whole region.
- **$L^*a^*b^*$ color difference of juncture.** This is defined as in Eq. (4), except that $(\overline{L^*}, \overline{a^*}, \overline{b^*})$ is the mean value of the $L^*a^*b^*$ values of the junctures between regions. When the color of one region gradually changes to the color of another region, and if only the $L^*a^*b^*$ color difference of the whole region is considered, the two regions may not be merged. To deal with this kind of situation, the $L^*a^*b^*$ color difference of

the juncture is necessary.

The following two steps are repeated until the merging requirement is met: (1) find the smallest $region_m$ and calculate the merging suitability between $region_m$ and its neighbor regions by using simplified fuzzy reasoning, and (2) merge $region_m$ with the region that has the highest suitability.

4. Determination of a Region's Importance

After segmentation, a region's features are calculated, and its importance is automatically determined through fuzzy reasoning. To reflect human knowledge, the reasoning rules are tuned through learning from a large number of data acquired in subjective assessment experiments. This section defines the features determining the importance of a region, lists the reasoning rules, and describes the acquisition of teaching data, the learning algorithm, the tuning of the rules, the reasoning, and the on-line learning.

4.1 Features Determining the Importance of a Region

Objects may be considered visually important if they stand out to the human eyes, or if they are meaningful or attractive to humans. Those parts that express the main subject of the image will be more important to humans than those parts that are not related to the subject [12]. According to visual psychology, brightness, color, shape, space perception, and movement perception are significant to the human visual system [13]. Whether a region is outstanding (pop-out [14]) to humans depends on the features of the region. We conducted an experiment on what features of a region determine its importance to human beings. The results showed that for an image with $Width$ pixels horizontally and $Height$ pixels vertically, and for $region_m$, the following features contribute to the importance:

- Area Ratio: The area ratio is defined as the ratio between the number of pixels in $region_m$ and the total number of pixels in the image, that is, the percentage of the entire image area occupied by $region_m$.
$$Arearatio_m = \frac{\Sigma pixel_m}{Width \times Height},$$
where $\Sigma pixel_m$ is the total number of pixels in $region_m$.
The area ratio denotes the relative size of $region_m$.
- Position: The position represents how far the center of gravity of $region_m$ is from the center of the image.
$$Position_m = \sqrt{(m_x - C_x)^2 + (m_y - C_y)^2},$$
where (C_x, C_y) are the coordinates of the image center, and (m_x, m_y) are the coordinates of the center of gravity of $region_m$.
- Compactness:
$$Compact_m = \frac{4\pi \times (area_m)}{(perimeter_m)^2},$$
where $area_m$ and $perimeter_m$ are respectively the area and perimeter of $region_m$. The compactness shows how compact $region_m$ is. It is equal to 1 when the region is round, and less when the boundary of the region is complicated.
- Border Connection:
$$Border_m = \frac{\Sigma connect_m}{2 \times (Width + Height)},$$
where $\Sigma connect_m$ is the number of pixels on the boundary of $region_m$ that are connected with the image border.
This identifies the possibility that a region is part of the background.
- Region Color: The mean value of each color component of CIE $L^*a^*b^*$ of the region i.e., $\overline{L^*}$, $\overline{a^*}$, $\overline{b^*}$, is used separately.
- Outstandingness: The outstandingness describes how outstanding a region is as compared with its neighboring regions. Usually, a region is more outstanding if there is a bigger color difference between it and its neighbor regions, or if it has larger and closer neighbor regions.
$$Outstand_m = \sum_{k=1, k\neq m}^{n} ||color_m - color_k||^2 \times (1 - distance_{mk}) \times Arearatio_k,$$
where
$$distance_{mk} = \frac{\sqrt{(m_x - k_x)^2 + (m_y - k_y)^2}}{\sqrt{Width^2 + \mathrm{Height}^2}}$$
and $||color_m - color_k||^2$ is defined as Eq. (4), while $Arearatio_k$ is the area ratio of $region_k$. The (m_x, m_y) and (k_x, k_y) are respectively the coordinates of the centers of gravity of $region_m$ and $region_k$.

4.2 Reasoning Rules

The following are the reasoning rules used in our scheme:

rule i:

$$
\begin{array}{llcl}
\mathit{IF} & \text{Arearatio} & \mathit{IS} & A_a \\
 & \text{Position} & \mathit{IS} & B_b \\
 & \text{Compact} & \mathit{IS} & C_c \\
 & \text{Border} & \mathit{IS} & D_d \\
 & L^* & \mathit{IS} & E_e \\
 & a^* & \mathit{IS} & F_f \\
 & b^* & \mathit{IS} & G_g \\
 & \text{Outstand} & \mathit{IS} & H_h \\
\mathit{THEN} & \text{Importance}_i & = & c_i,
\end{array}
\qquad (5)
$$

where

$$
\begin{array}{ll}
a,\ b,\ c,\ e,\ h & = \text{small, medium, large} \\
d & = \text{isborder, notborder} \\
f,\ g & = \text{color1, color2, color3,} \\
 & \quad \text{color4, color5} \\
i & = 1, 2, \cdots, l \\
 & \quad (l = 3^5 \times 2 \times 5^2 = 12150)
\end{array}
$$

i is the rule number, and l is the total number of rules. A_a, B_b, $\cdots$, H_h are the fuzzy sets of input variables. Each input variable is assigned a certain number of fuzzy sets. While most of them are assigned 3 fuzzy sets — "small," "medium," and "large"; the border connection is assigned only 2 sets — "isborder" and "notborder"; and the colors a^* and b^* are assigned 5 sets — "Color1," "Color2," "Color3," "Color4," and "Color5"; because they cannot be adequately represented by 3 sets. The $importance_i$ is the region importance of $rule_i$.

We infer the degree based on each rule by using Eq. (6), calculate the weighted sum, and obtain the region importance as the final reasoning result, using Eq. (7).

$$
\begin{aligned}
w_i = \ & \mu_{A_a}(Arearatio) \\
& \cap\ \mu_{B_b}(Position) \\
& \cap\ \mu_{C_c}(Compact) \\
& \cap\ \mu_{D_d}(Border) \\
& \cap\ \mu_{E_e}(L^*) \\
& \cap\ \mu_{F_f}(a^*) \\
& \cap\ \mu_{G_g}(b^*) \\
& \cap\ \mu_{H_h}(Outstand)
\end{aligned}
\qquad (6)
$$

$$
importance = \sum_{i=1}^{l} w_i c_i \Big/ \sum_{i=1}^{l} \omega_i \qquad (7)
$$

4.3 Acquisition of Teaching Data

Fuzzy reasoning is used to determine the importance of regions using the above 8 features as input. Although fuzzy logic can encode expert knowledge directly and easily using rules with linguistic labels, it usually takes a lot of time to design and tune the membership functions that quantitatively define these linguistic labels. Because of the large number of reasoning rules (12150) used in our scheme, tuning them is a tough task, and therefore an automatic tuning scheme is employed. Neural network learning techniques can automate this process and substantially reduce the development time and cost while improving the performance [15].

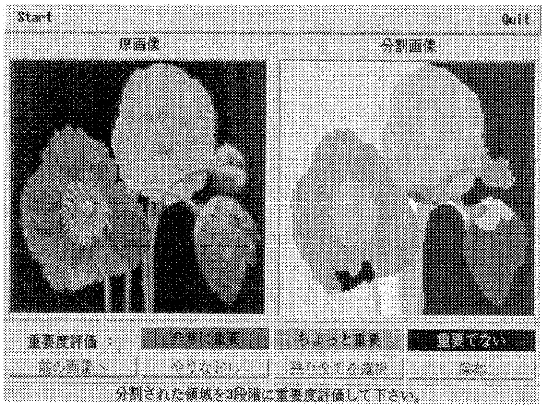


Fig. 2 Experimental tool.

To tune the reasoning rules by learning, it is essential to obtain input/output data that show under what region features the region is important. Therefore, we carried out an experiment with a group of university students.

The tool window shown in **Fig. 2** was displayed to the subjects on a computer screen. The original image is shown in the left of the window, while a segmented image of the original using the scheme described in Section 3 is shown in the right of the window. The segmented regions are indicated by different colors. The subjects looked at the original image and evaluated the importance of each region by assigning one of three levels: "very important," "somewhat important," and "not important."

The experiment was carried out with 15 subjects and 20 frames of images, and experimental data were acquired. The data from the 15 subjects were then averaged and normalized in the range [0, 1], to form the integral teaching data set.

4.4 Learning Algorithm

We chose the automatic tuning method proposed by Nomura [16], which is faster and of greater relevance to our approach than other methods. The premise membership function A_{ij} of $rule_i$ in our scheme is an isosceles triangle, with a base center value of a_{ij} and a base width of b_{ij}, shown in **Fig. 3** and defined as Eq. (8).

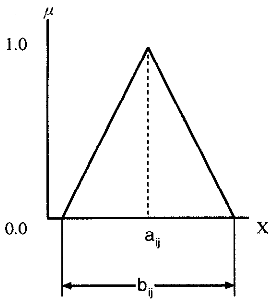


Fig. 3 Membership function A_{ij} of the premise.

$$A_{ij}(x_j) = \begin{cases} 1 - \dfrac{2 \times |x_j - a_{ij}|}{b_{ij}}; \\ \quad x_j \in \left(a_{ij} - \dfrac{b_{ij}}{2}, a_{ij} + \dfrac{b_{ij}}{2}\right) \\ 0; \text{ otherwise,} \end{cases} \tag{8}$$

where $b_{ij} > 0$. In our scheme a_{ij}, b_{ij}, and c_i, are the parameters to be tuned.

The evaluation function E is defined as follows:

$$E = \frac{1}{2}(y - y^r)^2, \tag{9}$$

where y^r is the conclusion of the teaching data, while y is the reasoned conclusion.

The parameters are modified with the aim of reducing E until the optimized goal is reached. The learning equations are shown as follows:

$$\begin{aligned} a_{ij}(t+1) &= a_{ij}(t) - K_a \times \frac{\partial E}{\partial a_{ij}}, \\ b_{ij}(t+1) &= b_{ij}(t) - K_b \times \frac{\partial E}{\partial b_{ij}}, \\ c_i(t+1) &= c_i(t) - K_a \times \frac{\partial E}{\partial c_i}, \end{aligned} \tag{10}$$

where K_a, K_b, K_c are the coefficients for learning, and t is the number of times learning occurs. By repeating Eq. (10), the learning procedure gives the optimized minimization of E.

4.5 Automatic Tuning of Fuzzy Reasoning Rules

The center value a_{ij} and base width b_{ij} of the triangular membership function, along with the real value c_i of the conclusion, are the parameters to be tuned by using the Delta-rule of the neural network learning procedure. The Delta-rule is a procedure for obtaining the optimized conclusion by repeating progressive learning, using given input and output data.

The learning in our scheme repeats the following two stages for the teaching data set until the optimized end condition is met:

(1) Do fuzzy reasoning, infer a conclusion from the given inputs, compare the inferred conclusion with the given output, and modify the conclusion importance c_i to reduce the inference error.

(2) Using the modified rules, repeat fuzzy reasoning, compare the inferred conclusion with the given output, and modify the parameters a_{ij}, b_{ij} of the membership function to reduce the inference error.

The fuzzy reasoning for determining the importance eventually results in an "MDU-Map," which defines the importance level of each MDU in the image (see Section 5.1).

4.6 On-line Learning

The scheme is capable of reasoning efficiently with an acceptable reasoning error, but differences in personal judgment must be considered. Therefore, an on-line learning ability is supported in our scheme to further enhance its practicality. The user is permitted to indicate the importance of each region, overriding the reasoning result given by the system, and decide whether the reasoning system will be taught or not. The on-line learning scheme is realized just like the off-line learning scheme described above. If the user decides to teach the new indications, the reasoning rules will be tuned on-line to adapt to his or her judgment, and future reasoning will be affected by the present teaching. If the user's indications are used only for the present image, the reasoning rules will not be tuned.

5. Region Representation

5.1 MDU-Map

Our scheme employs a data structure called an "MDU-Map", which is described by a two-dimensional array. Each of its element corresponds to one MDU and defines the importance level for all the blocks of the MDU, which is the smallest group of interleaved data units. The importance level can be any integer from 0 to 255, although the importance determination scheme we currently use only produces levels from 0 to 9. The larger the number, the higher the region importance. In the MDU-Map, 0 means that the MDU is the least important, while 9 means it is the most important. At the codec stage, in the same MDU, all the components, including the luminance and chrominance, are assigned the same quality. The MDU-Map is MDUHigh high and MDUWide wide, where MDUHigh and MDUWide are the height and width of an image in terms of the MDU.

In our scheme, there is another data structure called a "Scale Table." By consulting the table, the encoder and decoder can obtain the right quantization scale by using the importance level as an index.

5.2 Freeman Chain Codes

It becomes necessary in our scheme to code not only the region content but also descriptions of region contours. Among many representation methods, chain codes are most commonly used, because of their minimal storage requirements, good curvature description, and simplicity [18]. In our scheme, therefore, a region is represented as a boundary described by using Freeman chain codes and a scale. A region's boundary representation begins at the leftmost tangent point between the upper region boundary and its circumscribing rectangle. The boundary is then coded counter-clockwise. We treat all connected points of the same scale as one region, whatever its shape, in order to reduce the stream size.

5.3 Region Description Syntax

Although region description is performed in the encoding phase, it is necessary for the decoder to decode a compressed stream. Therefore, a region description needs be coded in a compressed image data stream. To conform with the JPEG standard [1], we use an application data segment to store the region description, which is shown in **Fig. 4**.

The items in Fig. 4 are defined as follows:

APP_n: Application data marker

lp: Application data segment length

BND_i: ith region description

We place our region data (boundary description, *Scale*, etc.) in BND_i, and use *Scale* to scale the quantization table so that we can obtain a different quality. The format of the boundary description is shown in **Fig. 5**.

The marker and parameters for region description shown in Fig. 5, are defined as follows:

BND: Boundary description marker; marks the beginning of a boundary description

Level: Importance level of region

x_0: X-coordinate of starting point of boundary description

y_0: Y-coordinate of starting point of boundary description

L_b: Number of boundary MDUs

B_i: Freeman chain code

APP$_n$	LP	BND$_1$	………	BND$_n$
2	2			(bytes)

Fig. 4 Region description embedded in an application data segment.

BND	Scale	X$_0$	Y$_0$	Lb	B$_1$	………	B$_n$
16	8	8	8	16	3		(bits)

Fig. 5 Format of region description.

The total byte cost for region description is as follows:

$$T = 4 + \sum_{i=1}^{n} \left(7 + INT\left(\frac{3 \times L_{bi} - 1}{8}\right)\right) \tag{11}$$

where L_{bi} is 1 less than the perimeter of $region_i$ in terms of MDUs, and n is the total number of regions.

The "example" quantization table given in section K.1 of the JPEG specification [1] is used. We scale the quantization tables to adjust the image quality. The sample quantization table is used as-is for a scale of 50.

5.4 Region Filling

Filling the interior of a region when its contour is given is one of the most common problems in many applications of image processing [17]. We developed an efficient region-filling technique that is used in both the encoding and decoding phases of our scheme. In the encoder, it is used to erase a region after the boundary of the region has been transformed into chain codes. In the decoder, it is used to restore the MDU-Map from the boundary described by chain codes in the application data segment of the coded stream.

Many region-filling techniques [18)~20)] are based on the parity-checking technique, which uses the fact that a straight line intersects any simple closed contour an even number of times. If we know that the first point of the line lies outside the region, then we traverse it and decide which segments are in the interior by counting the number of intersections. If the number is odd, then the segment lies in the interior; otherwise it does not. Two problems arise in the use of this technique. First, it is possible to have points from two or more sides mapped onto the same pixel, and this produces an incorrect count of the number of intersections. Second, problems may be caused by lines that are tangent to the contour [20].

The proposed region-filling technique here can fill any arbitrarily shaped region in which the above two problems have been solved efficiently. We suppose that the contour is closed, that the surrounding region can be deformed to the interior of a square, and that this is guaranteed by the importance-determining procedure. The following concepts are used in the technique:

- Isolated point: A point is called isolated when the number of its 8 neighbors is 0.
- Arc: A boundary is called an arc when it is not an isolated point and the area it surrounds is 0.
- Vertex: A point or a section of a horizontal line is called a vertex when both its left and right neighbors are located either above or below it. A horizontal scan line will be tangent to the contour at a vertex.
- Multi-way conjunction: A point is called a multi-way conjunction when it is not on an arc and the contour passes though it twice or more.
- Simple contour: A closed curve is called a simple contour when it contains no arcs or multi-way conjunctions.
- Arbitrarily shaped contour: The segmentation results in an arbitrarily shaped contour, which may contains simple contours, arcs, and multi-way conjunctions.

To solve the first problem, we transform an arbitrarily shaped contour into simple contours, arcs, and isolated points, fill the simple contours by using the parity-checking technique, and paste the filled simple region together with others to obtain the whole filled region. To fill a simple closed contour, the contour is scanned horizontally line by line from top to bottom. The second problem can be easily solved by treating multiway conjunctions as joints of two simple contours, and treating vertices as intersections with 0 length.

6. Compression

The encoder goes through each block in the original image to compress it, and check whether it is important or not; if it is important, it checks what the *Scale* is for the quantization table by referring to the MDU-Map and Scale Table. Different levels of block are compressed by using different quantization scales, shown in **Fig. 6**.

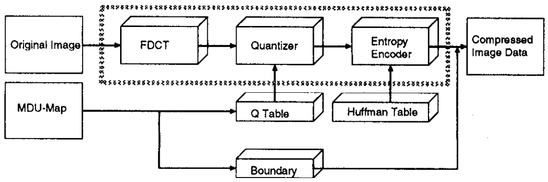


Fig. 6 Importance-adaptive compression.

It is the encoder's responsibility to translate an MDU-Map into a boundary description using chain codes, and to place it in the application data segment of the data stream. All important regions of an image are described as non-zero elements in the MDU-Map. The compression procedure is as follows:

- Write related tables into the stream.
- Make a backup for the MDU-Map, code regions' shape according to the MDU-Map, and do the following until the MDU-Map becomes completely zero (all important regions are coded):
  - Scan the MDU-Map horizontally from top to bottom, find the left uppermost non-zero element, take it as the starting-point for encoding the region's contour into chain codes, and put the chain codes and *Scale* into the JPEG stream.
  - Erase the coded region from the MDU-Map.
- Resume the MDU-Map, and code each block according to the MDU-Map.
  - FDCT.
  - Get *Scale* by referring to the MDU-Map and Scale Table.
  - Get $Q(u, v)$ by using Eq. (3).
  - Quantize using the above $Q(u, v)$ and Eq. (1).
  - Entropy-code.

The application data segment can be ignored by standard JPEG, so the stream coded by using our scheme is compatible with the syntax of standard JPEG.

7. Decompression

The decoder takes a compressed data stream as input, obtains a region boundary description from the application data segment, reproduces the region by region filling, does the same for

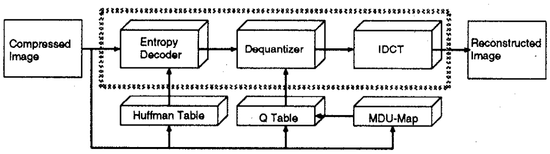


Fig. 7 Importance-adaptive decompression.

all the regions, and, through pasting, forms an MDU-Map that is exactly the same as the original one. By referring to the map, the decoder can decode each block accordingly, as shown in **Fig. 7**.

The decoding procedure is as follows:

- Initialize related tables by decoding the header of the stream.
- Restore regions' shapes by decoding the application data segment as follows until all *BND*s are decoded. The MDU-Map is initialized to zero.
  - Decode *BND* to obtain a region boundary (refer to Fig. 4 and Fig. 5).
  - Fill the above region by using the region-filling technique described in Section 5.4.
  - Paste the above filled region into the MDU-Map.
- Decode each block according to the MDU-Map.
  - Entropy-decode.
  - Get *Scale* by referring to the MDU-Map and Scale Table.
  - Get $Q(u, v)$ by using Eq. (3).
  - Dequantize, using the above $Q(u, v)$ and Eq. (2).
  - IDCT.

8. Experimental Results and Conclusion

The experimental data demonstrate that the reasoning rules' tuning procedure is convergent after a certain number of iterations of learning, as shown in **Fig. 8**, where the *Times* and *Error* are respectively the number of iterations of learning and the reasoning error. However, the learning speed depends on the learning coefficients, namely, K_a, K_b, and K_c. In Fig. 8, K_a, K_b, K_c are gradually decreased in step with the learning progress, in order to both accelerate and stabilize the learning. Figure 8 shows that after 2,500 iterations of learning, the reasoning error could reach 0.055, which is considered to be small enough for our compression application. The importance determination scheme is feasible for most images, and reflects the most common human sense of what is important in an image. In addition, the on-line learning capability is available, so the practicability of the scheme is further enhanced.

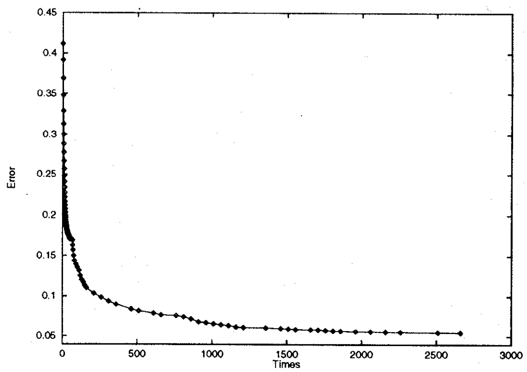


Fig. 8 Learning curve for teaching images.

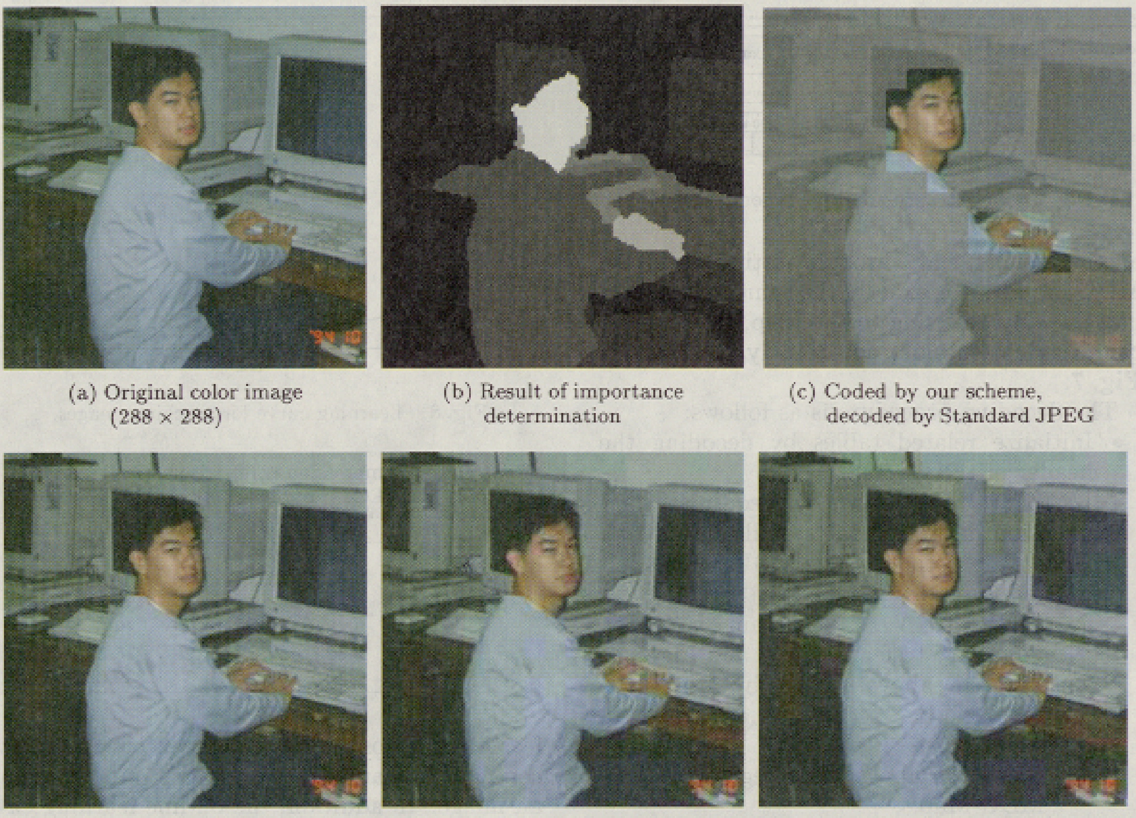
Table 1 Compression comparison with standard JPEG.

Image	*Size*	bpp_1	$PSNR_1$	bpp_2	$PSNR_2$	*Ratio*
1	786447	0.54	29.8	1.46	34.7	0.37
2	786447	0.68	28.3	1.80	32.8	0.38
3	248847	0.56	33.7	1.19	36.7	0.47
4	248847	0.48	34.5	1.02	39.6	0.47
5	248847	0.69	32.6	1.28	36.6	0.54
6	410535	0.59	33.2	1.34	39.6	0.44

We have used our importance adaptive scheme with six typical color images that were unknown (not used as teaching images). If we assign the most important regions the same quality as in standard JPEG, and the less important regions a lower quality, the size of the compressed image is reduced dramatically, and it still retains a satisfactory psychological quality. **Table 1** gives a qualitative result, since the size and PSNR of the compressed image vary with different combinations of scales. In Tables 1 and 2, subscripts 1 and 2 respectively designate values for the images output by our method, and the images output by the standard JPEG scheme. The *Size* denotes the ppm file size of each original image in bytes, while the *bpp* denotes number of bits per pixel in the compressed image. The *Ratio* is the ratio of bpp_1 to bpp_2, and the *PSNR* (Peak Signal-to-Noise Ratio) of $region_m$ is defined as:

$$PSNR_m = 10 \log_{10} \frac{255^2}{MSE_m} \quad \text{dB}.$$

Note that the mean square error for $region_m$ is defined as:



(a) Original color image (288 × 288)
(b) Result of importance determination
(c) Coded by our scheme, decoded by Standard JPEG
(d) Standard JPEG (0.75 bpp, 38.1 dB)
(e) Standard JPEG (0.46 bpp, 34.1 dB)
(f) Our scheme (0.46 bpp, 34.0 dB)

Fig. 9 Importance adaptive compression and decompression result.

$$MSE_m = \frac{1}{A_m} \sum_{(i,j)\in region_m} (Y_{ij} - \hat{Y_{ij}})^2,$$

where Y_{ij} and $\hat{Y_{ij}}$ denote the original and reconstructed luminance level (Y), respectively, and A_m denotes the area of $region_m$. The overall $PSNR$ of image can also be calculated by using the above equations.

To compare the performance of our scheme with that of the standard JPEG scheme at the same bit rate, we distributed a questionnaire to 15 university students. The above-mentioned 6 color images were respectively coded with nearly the same size by using our scheme and the standard JPEG scheme. The questionnaire result shows that our scheme gives a better subjective quality than the standard JPEG scheme, at the same bit rate. **Table 2** summarizes the questionnaire results.

The numbers given in the columns labeled *Worse*, *Same*, and *Better* in Table 2 respectively show the number of subjects who thought that the quality of the image output by our scheme was *Worse*, the *Same*, or *Better* than that output by the standard JPEG scheme.

Table 2 Questionnaire results.

Image	*bpp*	$PSNR_1$	$PSNR_2$	*Worse*	*Same*	*Better*
1	0.47	29.0	29.1	0	4	11
2	0.57	27.4	27.2	0	6	9
3	0.52	33.5	33.8	0	10	5
4	0.46	34.0	34.1	0	1	14
5	0.59	31.4	31.4	0	5	10
6	0.54	32.9	33.2	0	7	8

Figure 9 gives a demonstrative example, while **Table 3** gives the regional data for the images in Fig. 9. In Table 3, *No.* stands for category number; all the regions with the same importance level are classed into one category. I, A, C, S, B and P respectively stand for the importance level, the geometric area in terms of MDUs, the cost in bytes of the chain codes used for boundary representation, the quantization table scale, the cost in bytes of the entropy used for content coding, and the PSNR of the region category. The subscripts 1, 2, and 3 respectively designate the values for Fig. 9 (f), (e),

Table 3 Regional data of Fig. 9.

No.	1	2	3	4	5	6	7
I	9	7	5	4	3	2	0
A	19	11	11	79	10	70	124
C	12	11	11	30	10	40	0
S_1	50	80	110	125	140	180	200
S_2	125	125	125	125	125	125	125
S_3	50	50	50	50	50	50	50
B_1	625	303	252	1063	139	633	1041
B_2	354	223	227	1063	153	777	1368
B_3	625	407	393	1886	291	1275	2307
P_1	36.5	33.8	33.1	34.2	31.9	34.6	33.5
P_2	32.2	31.0	31.0	34.2	32.0	35.4	35.0
P_3	36.5	34.7	34.7	37.8	35.7	39.8	39.2

and (d).

We obtain a different quality by changing the *Scale* of Eq. (3). Figure 9 (b) shows the result of determining the importance of a region; the lighter the area, the more important the region, and correspondingly, the higher its importance level in Table 3. If we compare Fig. 9 (e) with Fig. 9 (f) and refer to Table 3, it is clear that, since the importance determination and bit allocation reflect the human sense of importance, we should use the same file size to produce a better reconstructed subjective quality by giving important regions higher quality (higher PSNR and bit rate) while giving unimportant ones lower quality (lower PSNR and bit rate). For example, the importance level of the face area is determined as 9, while that of the background is determined as 0. The face area consists of 19 MDUs, and needs 12 bytes for representation of its boundary. The quantization scale for the face region is 50 in Fig. 9 (f), 125 in (e), 50 in (d); therefore 625, 354, and 625 bytes are respectively needed to code its content, and the resulting PSNRs are respectively 36.5 dB, 32.2 dB, and 36.5 dB. The face area in Fig. 9 (f) is obviously clearer than that in (e). The face area in Fig. 9 (f) has the same quality as that in (d), but the file size is much smaller.

Our experimental results, together with subjective quality evaluation, demonstrate the attractive application prospects of the scheme. The scheme can be used to allocate higher quality only to the important regions, to reduce file size by further compressing the unimportant parts, and to ensure the best visual quality for a given compression ratio. Moreover, the importance-adaptive scheme guarantees optimized importance allocation.

In addition, the proposed adaptive JPEG codec scheme can be used in progressive JPEG mode or for scrambling JPEG images, while the importance adaptive idea can be easily adopted in MPEG coding, where Freeman chain codes are not needed because there is already a per-macroblock quantization scale in MPEG syntax for macroblock quality or rate control [21].

References

1) ISO/IEC 10918-1, Information Technology – Digital Compression and Coding of Continuous-tone Still Images: Requirements and Guidelines (1994).

2) Wallace, G.: The JPEG Still Picture Compression Standard, *Comm. ACM*, Vol.34, No.4, pp.31–44 (1991).

3) Crouse, M. and Ramchandran, K.: Joint Thresholding and Quantizer Selection for Decoder-compatible Baseline JPEG, *Proc. ICASSP*, pp.2331–2334 (1995).

4) Chen, W. and Pratt, W.: Scene Adaptive Coder, *IEEE Trans. Comm.*, Vol.COM-32, No.3, pp.225–232 (1984).

5) Wu, X. and Fang, Y.: A Segmentation-based Predictive Multiresolution Image Coder, *IEEE Trans. Image Processing*, Vol.4, No.1, pp.34–47 (1995).

6) Salari, E. and Lin, S.: Segmentation Based Coding Algorithm for Low Bit-rate Video, *IEEE ICASSP*, pp.2241–2244 (1995).

7) Chen, H., Civanlar, M. and Haskell, B.: A Block Transform Coder for Arbitrarily Shaped Image Segments, *Proc. IEEE ICIP '94*, pp.85–89 (1994).

8) Ito, N., Kamekura, R., Shimazu, Y., Yokoyama, T. and Matsushita, Y.: The Combination of Edge Detection and Region Extraction in Non-parametric Color Image Segmentation, Information Sciences, Vol.92, No.1-4, pp.277–294 (1996).

9) Ito, N., Shimazu, Y., Yokoyama, T. and Matsushita, Y.: Fuzzy Logic Based Non-parametric Color Image Segmentation with Optional Block Processing, *Proc. 23rd ACM Computer Science Conference*, pp.119–126 (1995).

10) Zhao, J., Shimazu, Y., Ohta, K., Hayasaka, R. and Matsushita, Y.: An Outstandingness Oriented Image Segmentation and Its Application, *Proc. ISSPA '96*, pp.45–48 (1996).

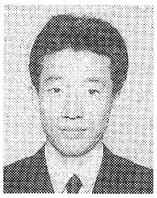
11) Khotanzad, A. and Bouarfa, A.: Image Segmentation by a Parallel, Non-parametric Histogram Based Clustering Algorithm, *Pattern Recogn.*, Vol.23, No.9, pp.961–973 (1990).

12) Nishiyama, H. and Matsushita, Y.: An Image Retrieval System Considering Image Composition, *Trans. IPSJ*, Vol.37, No.1, pp.101–109 (1996).

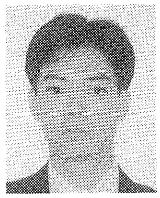
13) Tasaki, K., Ohyama, T. and Hiwatashi, K.:

Theory of Color Cognition, Asakura Press (1979).
14) Davidoff, J.: *Cognition through Color*, MIT Press (1991).
15) Asakawa, K. and Takagi, H.: Neural Networks in Japan, *Comm. ACM*, Vol.37, No.3, pp.106–112 (1994).
16) Nomura, H., Hayashi, I. and Wakami, N.: A Self-tuning Method of Fuzzy Reasoning by Delta Rule and Its Application to a Moving Obstacle Avoidance, *J. of Japan Society for Fuzzy Theory and Systems*, Vol.4, No.2, pp.379–388 (1992).
17) Ali, S. and Burge, R.: A New Algorithm for Extracting the Interior of Bounded Regions Based on Chain Coding, *Computer Vision, Graphics and Image Processing*, Vol.43, pp.256–264 (1988).
18) Grzeszczuk, R., Eddins, S. and Defanti, T.: Filling Regions Using a Crack Code Boundary Description, *SPIE*, Vol.1778, pp.226–237 (1992).
19) Chang, L. and Leu, K.: A Fast Algorithm for the Restoration of Images Based on Chain Codes Description and Its Applications, *Computer Vision, Graphics, and Image Processing*, Vol.50, pp.296–307 (1990).
20) Cai, Z.: Restoration of Binary Images Using Contour Direction Chain Codes Description, *Computer Vision, Graphics, and Image Processing*, Vol.41, pp.101–106 (1988).
21) ISO/IEC 11172-2, Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s, Part 2: Video (1993).

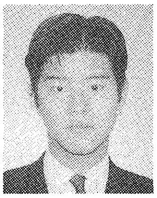
(Received April 5, 1996)
(Accepted May 8, 1997)



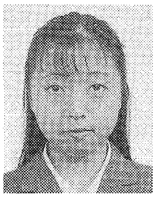
Jiying Zhao was born in 1963. He received his B.E., M.E. in Computer Engineering, and Ph.D. degrees in Electrical Engineering from North China Electric Power University respectively in 1983, 1988, and 1992. He had worked for the same university as an assistant, lecturer, and associate professor during the period from 1983 to 1994. He is currently a Ph.D. student again, with the Department of Instrumentation Engineering, Faculty of Science and Technology, Keio University. His current research interests are in digital image processing and coding. He is a Student Member of Information Processing Society of Japan.



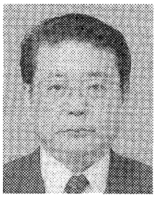
Yoshihisa Shimazu was born in 1971. He received his B.E., M.E. degrees from Keio University respectively in 1994, 1996. He is with Matsushita Electric Industrial Co. Ltd. His current research interests are in image processing.



Koji Ohta was born in 1972. He received his B.E., M.E. degrees from Keio University respectively in 1995, 1997. He is with Sony Corporation. His current research interests are in image processing.



Rina Hayasaka was born in 1974. She received her B.E. degree from Keio University in 1996. She is currently M.E. student with the same university. Her current research interests are in image processing. She is a Student Member of Information Processing Society of Japan.



Yutaka Matsushita was born in 1939. He received his B.E. in Electrical Engineering from Keio University, M.S. in Computer Science from University of Illinois and Ph.D. in Electrical Engineering from Keio University, respectively in 1963, 1969 and 1978. From 1963 to 1988, He joined OKI Electric Industry Ltd. where he had served as the director of the integrated systems division. Since 1989, He has been a professor in Department of Instrumentation Engineering of Keio University. He is an author or co-author of 17 books and published over 100 papers. He received best author award from IPSJ in 1993. His more recent works have been in the area of groupware, mobile networks and telecommunication architecture. He is currently serving as the president of SIG GW of IPSJ and SIG MIS of IEICE. He is a member of IEEE, ACM, IEICE, Japanese Society for Artificial Intelligence and Japanese Society for Fuzzy Theory and Systems.