Color image segmentation using competitive learning

Article *in* IEEE Transactions on Pattern Analysis and Machine Intelligence · January 1995

DOI: 10.1109/34.387488 · Source: IEEE Xplore

Michael A Arbib
University of California, San Diego

Color Image Segmentation Using Competitive Learning

Toshio Uchiyama and Michael A. Arbib, *Member, IEEE*

***Abstract*— This paper presents a color image segmentation method which divides color space into clusters. Competitive learning is used as a tool for clustering color space based on the least sum of squares criterion. We show that competitive learning converges to approximate the optimum solution based on this criterion theoretically and experimentally. We applied this method to various color scenes and show its efficiency as a color image segmentation method. We also show the effects of using different color coordinates to be clustered by some experimental results.**

I. INTRODUCTION

SINCE computer performance has improved in recent years, it has become easier to deal with the large amounts of data in color images. However, much work in computer vision still focuses on gray scale images, even though techniques for gray scale images usually cannot be applied to color images. For example, segmentation of a color image is a more complicated problem than that of a gray scale image.

Many image segmentation techniques have been proposed [1]–[3]. These methods could be categorized into methods for dividing an image space and those for clustering a feature space derived from an image. Region growing [4] and merge-split [5] are the former methods. The segmentation method that we use here belongs to the latter category. Each pixel is assigned to a color index of the cluster to which it belongs. This paper demonstrates mathematical properties of applying the latter technique to image segmentation so that one can use it efficiently.

Multiple histogram-based thresholding schemes [6], [7], [25] are proposed to divide multidimensional data such as a color image. This method divides color space by histogram thresholding of each axis. Since thresholding is a technique for gray scale images, there are limitations when dividing multiple dimensions. Fig. 1(a) shows the 2-D case of thresholding. As the figure shows, the shape of the cluster is a rectangle. It is not appropriate to divide multidimensional data by thresholding 1-D axes, because there is no criterion for multidimensional data in these schemes.

Fig. 1(b) is the Voronoi tessellation in which each position of space belongs to the nearest weight vector $\boldsymbol{W}_i$. This division is vector quantization by weight vector $\boldsymbol{W}_i, (i = 1, \cdots, n)$,

Manuscript received June 30, 1993; revised June 8, 1994. Reccommended for acceptance by Associate Editor R. Bolle.

T. Uchiyama is with NTT DATA Communications Systems Corporation, 66-2 Horikawa-cho Saiwai-ku Kawasaki, 210 Japan.

M. A. Arbib is with the Center for Neural Engineering, University of Southern California, Los Angeles, CA 90089-2520.

IEEE Log Number 9406175.

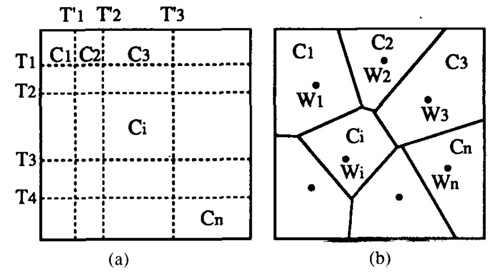


Fig. 1. Division of 2-D data. (a) Multiple histogram-based thresholding. (b) Vector quantization (= clustering).

and is more flexible to the shape and arrangement of clusters. If quantization error is minimum in all possible $\boldsymbol{W}_i$, this division becomes clustering based on the least sum of squares criterion (ref. next section). Clustering based on the criterion is unique and reasonable in the sense of the least sum of squares. We will present a method that realizes this clustering. This method does not require any human interaction, while thresholding schemes sometimes do.

Among the clustering techniques based on the least sum of squares criterion, the K-means (C-means) [11] algorithm and ISODATA [12], which added some parameters to the K-means algorithm, are famous and used for clustering of multispectral data [13], [14]. It was shown that the K-means algorithm converges to a local minimum solution, but not necessarily to the optimum solution. Hence, it is a common problem of clustering techniques to get a good approximate solution. Several improvements of clustering techniques have been proposed [15], [16]. Here, we use competitive learning for the clustering, with additional mechanisms to improve performance of this.

Since the shapes and distribution of clusters depend on the color space to be divided, the selection of color space is important [25]. When a color image is presented by RGB, selection of the color space is a selection of a transformation. This paper presents the mathematical feature of linear transformation before clustering. In experiments on color image segmentation, we also use $L{*}u{*}v{*}$ uniform color space.

The next section shows the relationship between clustering and vector quantization. The third section shows the competitive learning algorithm. We show theoretically and experimentally that competitive learning converges to a good approximate solution of the clustering. The fourth section describes transformations of color space and color image segmentation results.

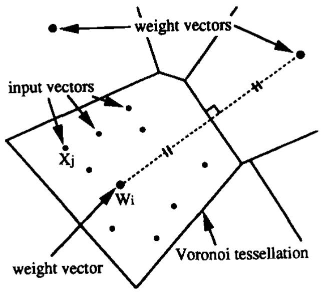


Fig. 2. Vector quantization.

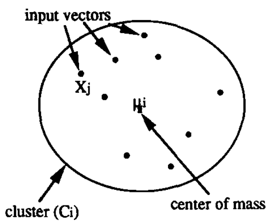


Fig. 3. Clustering.

II. Clustering and Vector Quantization

Clustering based on the least sum of squares criterion gives the most separable clustering. We show the definition and mathematical features of it.

Vector quantization means mapping input vectors to a finite number of weight vectors and requires us to decide a set of weight vectors, while clustering requires us to decide a partitioning of input vectors. These two problems are different, but there is a close relationship between them. Therefore, clustering can be replaced by vector quantization. We will explain the details in Sections II-A and II-B.

A. Mathematical Expression

This subsection shows the definitions of clustering based on the least sum of squares criterion and vector quantization.

Let $\boldsymbol{X}_j$ $(j = 1, \cdots, m)$, C_i $(i = 1, \cdots, n)$, $\boldsymbol{W}_i$ $(i = 1, \cdots, n)$ $(m > n)$ be a finite number of input vectors $(=\mathcal{X})$, clusters $(=\mathcal{C})$, and weight vectors $(=\mathcal{W})$, respectively. Fig. 2 and Fig. 3 show vector quantization and clustering.

We define a partition function $\mathcal{C}_f(\mathcal{W})$, that is a Voronoi tessellation determined by weight vector $\mathcal{W}$ in Fig. 2, as

$$\boldsymbol{X}_j \in C_i \text{ iff } \forall k (k \neq i) \|\boldsymbol{X}_j - \boldsymbol{W}_i\|^2 < \|\boldsymbol{X}_j - \boldsymbol{W}_k\|^2, \quad (1)$$

and define a weight vector function $\mathcal{W}_f(\mathcal{C})$, that is center of mass in cluster C_i, as

$$\vec{\mu}_i \equiv \frac{\sum_{\boldsymbol{X}_j \in C_i} p_j \boldsymbol{X}_j}{\sum_{\boldsymbol{X}_j \in C_i} p_j}, \quad (2)$$

$$\mathcal{W}_f(\mathcal{C}) = \{\vec{\mu}_i \mid i = 1, \cdots, n\} \quad (3)$$

where p_j is the probability of $\boldsymbol{X}_j$. In this paper $p_j = 1/m$.

When expected squared-error distortion is defined by

$$D(\mathcal{C}, \mathcal{W}) = \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j \|\boldsymbol{X}_j - \boldsymbol{W}_i\|^2, \quad (4)$$

we can define "Clustering based on the least sum of squares" and "Vector quantization" as follows.

We use the word "length" to denote the number of C_i in $\mathcal{C}$.

Definition of clustering: When $\mathcal{X}$ and a length of clusters n are given, the clustering based on the least sum of squares is the set of clusters $\mathcal{C}$ that minimizes the within-cluster sum of squares S_W ("$_W$" stands for "within-cluster") defined by

$$S_W(\mathcal{C}) \equiv D(\mathcal{C}, \mathcal{W}_f(\mathcal{C})) = \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j \|\boldsymbol{X}_j - \vec{\mu}_i\|^2. \quad (5)$$

Definition of vector quantization: When $\mathcal{X}$ and a length of weight vectors n are given, choose $\mathcal{W}$ to quantize $\mathcal{X}$. Vector quantization is evaluated by the quantization error defined by

$$E(\mathcal{W}) \equiv D(\mathcal{C}_f(\mathcal{W}), \mathcal{W}) = \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j \|\boldsymbol{X}_j - \boldsymbol{W}_i\|^2. \quad (6)$$

The motivation for using the least within-cluster sum of squares criterion for clustering is the following. We define total sum of squares S_T and between-cluster sum of squares S_B by

$$S_T \equiv \sum_{i=1}^{n} \sum_{j=1}^{m} p_j \|\boldsymbol{X}_j - \vec{\mu}_0\|^2 \quad (7)$$

$$S_B \equiv \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j \|\vec{\mu}_i - \vec{\mu}_0\|^2 \quad (8)$$

where

$$\vec{\mu}_0 \equiv \frac{\sum_{j=1}^{m} p_j \boldsymbol{X}_j}{\sum_{j=1}^{m} p_j} \quad (9)$$

is the global center of mass of $\mathcal{X}$, and we have

$$S_T = S_W + S_B. \quad (10)$$

Since S_T is independent of $\mathcal{C}, \mathcal{W}$, minimization of S_W is equivalent to maximization of S_B and in this sense the least sum of squares criterion gives the most separable clustering [18].

The number $S(m, n)$ of possible partitioning $\mathcal{C}$ of length n of the m input vectors $\mathcal{X}$ is given by

$$S(m, n) = \frac{1}{n!} \sum_{i=1}^{n} (-1)^{n-i} \binom{n}{i} i^m \quad (11)$$

(see [10]). It is difficult to obtain an optimum solution by searching all the enumeration in a reasonable computation time. Hence, a technique for restricting the search by vector quantization technique is useful. The next subsection shows how the problem of selecting the clustering $\mathcal{C}$ becomes the problem of selecting $\mathcal{W}$. Therefore, clustering can be replaced by vector quantization.

B. Theories

As we showed in the previous subsection, the goal of clustering is to select a partitioning $\mathcal{C}$ that minimizes S_W. However, the selection of $\mathcal{C}$ can be replaced by that of $\mathcal{W}$ from Theorem 1. Here we present the theorem and prove it.

Lemma 1 [17]: When $\mathcal{C}$ is given, the $\mathcal{W}$ determined by $\boldsymbol{W}_i = \vec{\mu}_i$ minimizes $D(\mathcal{C},\mathcal{W})$ and makes $D(\mathcal{C},\mathcal{W})$ equal to $S_W(\mathcal{C})$.

$$\min_{\mathcal{W}} D(\mathcal{C},\mathcal{W}) = S_W(\mathcal{C}) = D(\mathcal{C},\mathcal{W}_f(\mathcal{C})). \tag{12}$$

Proof:

$$\begin{aligned}
&D(\mathcal{C},\mathcal{W}) - S_W(\mathcal{C})\\
&= \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j(\|\boldsymbol{X}_j - \boldsymbol{W}_i\|^2 - \|\boldsymbol{X}_j - \vec{\mu}_i\|^2)\\
&= \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j(\boldsymbol{X}_j^2 - 2\boldsymbol{X}_j\boldsymbol{W}_i + \boldsymbol{W}_i^2 - \boldsymbol{X}_j^2 + 2\boldsymbol{X}_j\vec{\mu}_i - \vec{\mu}_i^2)\\
&= \sum_{i=1}^{n} \left[\left(\sum_{\boldsymbol{X}_j \in C_i} p_j \boldsymbol{X}_j \right)(-2\boldsymbol{W}_i + 2\vec{\mu}_i) + \sum_{\boldsymbol{X}_j \in C_i} p_j(\mathbf{W}_i^2 - \vec{\mu}_i^2) \right]\\
&= \sum_{i=1}^{n} \left[\left(\vec{\mu}_i \sum_{\boldsymbol{X}_j \in C_i} p_j \right)(-2\boldsymbol{W}_i + 2\vec{\mu}_i) + \sum_{\boldsymbol{X}_j \in C_i} p_j(\boldsymbol{W}_i^2 - \vec{\mu}_i^2) \right]\\
&= \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j(\boldsymbol{W}_i^2 - 2\boldsymbol{W}_i\vec{\mu}_i + \vec{\mu}_i^2)\\
&= \sum_{i=1}^{n} \sum_{\boldsymbol{X}_j \in C_i} p_j\|\boldsymbol{W}_i - \vec{\mu}_i\|^2 \geq 0.
\end{aligned} \tag{13}$$

We note that Lemma 1 is related to minimizing expected squared-error distortion [9].

$$(S_W = \min_{\mathcal{C}} S_W(\mathcal{C}) \wedge \boldsymbol{W}_i = \vec{\mu}_i) \text{ iff } D = \min_{\mathcal{C},\mathcal{W}} D(\mathcal{C},\mathcal{W}). \tag{14}$$

The equation (14) shows that the least sum of squares criterion is equivalent to the least expected squared-error distortion criterion in clustering problems.

Lemma 2 [17]: When $\mathcal{W}$ is given, the $\mathcal{C}$ determined by the Voronoi tessellated division of $\mathcal{W}$ minimizes $D(\mathcal{C},\mathcal{W})$, and makes $D(\mathcal{C},\mathcal{W})$ equal to $E(\mathcal{W})$

$$\min_{\mathcal{C}} D(\mathcal{C},\mathcal{W}) = E(\mathcal{W}) = D(\mathcal{C}_f(\mathcal{W}),\mathcal{W}). \tag{15}$$

Theorem 1 [17]: If $\mathcal{C}_0$ minimizes S_W, $\mathcal{C}_0$ is determined by a Voronoi tessellated division of $\mathcal{W}$ that satisfies $\boldsymbol{W}_i = \vec{\mu}_i$.

$$S_W(\mathcal{C}_0) = \min_{\mathcal{C}} S_W(\mathcal{C}) \tag{16}$$

$$\Longrightarrow \boldsymbol{W}_i = \vec{\mu}_i \wedge \mathbf{X}_j \in C_{0i} \text{ iff } \forall k(k \neq i)\|\boldsymbol{X}_j - \boldsymbol{W}_i\|^2 < \|\boldsymbol{X}_j - \boldsymbol{W}_k\|^2. \tag{17}$$

Proof: There exists $\mathcal{C}_0$ that minimizes S_W. We define $\mathcal{W}_0 = \mathcal{W}_f(\mathcal{C}_0)$. From (5) we have

$$D(\mathcal{C}_0,\mathcal{W}_0) = S_W. \tag{18}$$

Then we define $\mathcal{C}' = \mathcal{C}_f(\mathcal{W}_0)$. From Lemma 2

$$D(\mathcal{C}',\mathcal{W}_0) \leq D(\mathcal{C}_0,\mathcal{W}_0)(= \text{ iff } \mathcal{C}_0 = \mathcal{C}'). \tag{19}$$

Then we define $\mathcal{W}' = \mathcal{W}_f(\mathcal{C}')$. From Lemma 1 we have

$$D(\mathcal{C}',\mathcal{W}') \leq D(\mathcal{C}',\mathcal{W}_0)(= \text{ iff } \mathcal{W}_0 = \mathcal{W}'). \tag{20}$$

Since $D(\mathcal{C}_0,\mathcal{W}_0)$ is minimum from (18), $\mathcal{C}_0 = \mathcal{C}' = \mathcal{C}_f(\mathcal{W}_0)$ and $\mathcal{W}_0 = \mathcal{W}' = \mathcal{W}_f(\mathcal{C}_0)$.

III. Clustering by Competitive Learning

In this section we describe how competitive learning realizes clustering based on the least sum of squares criterion.

Competitive learning converges to an equilibrium that is regarded as the solution of the vector quantization problem (ref. Section III-B). Since clustering can be replaced by vector quantization (ref. Section II), competitive learning can realize clustering. We experimentally show the reality of this in Section III-C.

A. Algorithm

We present the competitive learning algorithm that we use in this paper. Since competitive learning has the problem that a few units may monopolize input vectors $\mathcal{X}$ so that no more competition occurs [19], we add a rule that distributes units to adapt them for $\mathcal{X}$ to prevent the monopoly problem. This rule gradually generates units until the number of units reaches n, the maximum allowed number of clusters. Initially, only one unit exists. The unit distribution rule is presented in 2 c) in *Algorithm.* In *Algorithm*, N_R is the number of repetitions, $N_{\max}$ is the maximum number of repetitions, θ_t is the threshold of times and γ is the learning rate of competitive learning.

Algorithm

1) Initialization

   One unit is set up, with weight vector $\boldsymbol{W}$ equal $\vec{\mu}_0$. It has a variable *wincount* that is initialized to 0 and that shows how many times each unit wins. We initialize the number of repetitions N_R to 0.

2) Competitive learning

   a) Select one input vector $\boldsymbol{X}$ randomly from $\mathcal{X}$. Decide a winner derived from the input vector $\boldsymbol{X}$. A unit that is closest to $\boldsymbol{X}$ in the squared Euclidean distance $\|\boldsymbol{X} - \boldsymbol{W}_i\|^2$ is chosen as a winner $\boldsymbol{W}_{\text{winner}}$. (If there are several such units, one is chosen at random.)

b) Update the weight vector of winner $\boldsymbol{W}_{\text{winner}}$ by

$$\Delta \boldsymbol{W}_{\text{winner}} = \gamma(\boldsymbol{X} - \boldsymbol{W}_{\text{winner}}). \tag{21}$$

Add 1 to $wincount$ of winner.

c) If a winner's $wincount = \theta_t$ and (number of units $< n$), then generate a new unit with the same $\boldsymbol{W}$ as the winner and clear the $wincount$ of both to 0.

d) Add 1 to N_R. If $(N_R = N_{\max})$, then stop competitive learning, else continue competitive learning from a).

3) Clusters C are determined by (1).

When a new unit is generated in 2 c), two units are located at the same place. If an $\boldsymbol{X}$ that is closest to these two units is input, one of the two units becomes the winner and updates its $\boldsymbol{W}$, while the other does not. Hence, the problem that two units move together does not occur.

We now introduce the rule added to competitive learning. This rule generates units where the density of input vectors is high. Since an inappropriate initialization of units causes the problem of monopoly, this rule improves the performance of competitive learning. The increase of θ_t gives better accuracy in the distribution, but it requires a larger maximum number of repetitions $N_{\max}$. When we decide the values of θ_t and $N_{\max}$, they must satisfy the following necessary condition. Considering the fastest and the slowest cases of generating N_U units, we have

$$(N_U - 1)\theta_t \le N_R < (2N_U - 1)\theta_t. \tag{22}$$

When the nth unit is generated in the slowest case,

$$N_R = ((2(n-1) - 1)\theta_t = (2n-3)\theta_t. \tag{23}$$

Therefore, the $N_{\max}$ that guarantees to generate the n units satisfies

$$N_{\max} \ge (2n-3)\theta_t. \tag{24}$$

This is only a necessary condition. We show a condition which yields a good approximate solution, as shown through experimentation, in Section III-C.

B. Proofs

We mention about a convergence of competitive learning. The following Theorem 2 is closely related to the *AVQ Centroid Theorem* of Kosko [21].

Theorem 2: When $\mathcal{W}$ satisfies $\boldsymbol{W}_i = \vec{\mu}_i$, competitive learning is at an equilibrium state.

Proof: From $\boldsymbol{W}_i = \vec{\mu}_i$ and (2),

$$\boldsymbol{W}_i = \frac{\sum_{\boldsymbol{X}_j \in C_i} p_j \boldsymbol{X}_j}{\sum_{\boldsymbol{X}_j \in C_i} p_j}. \tag{25}$$

Then we have

$$\sum_{\boldsymbol{X}_j \in C_i} p_j \boldsymbol{X}_j = \sum_{\boldsymbol{X}_j \in C_i} p_j \boldsymbol{W}_i, \tag{26}$$

$$0 = \sum_{\boldsymbol{X}_j \in C_i} p_j(\boldsymbol{X}_j - \boldsymbol{W}_i). \tag{27}$$

Multiplying by γ, we have

$$0 = \sum_{\boldsymbol{X}_j \in C_i} p_j \gamma(\boldsymbol{X}_j - \boldsymbol{W}_i). \tag{28}$$

Competitive learning is at equilibrium, because (28) means that the expected movement equals 0. Q.E.D.

From both Theorems 1 and 2, the $\mathcal{W}$ obtained by competitive learning satisfies the necessary condition of clustering based on the least sum of squares. In other words, competitive learning reduces the number of possible solutions without excluding the optimum solution. It is important that competitive learning converges to a local optimum solution of the clustering problem as well as that of the vector quantization problem which minimizes the quantization error E [22].

C. Clustering in Gray Scale Images

We will show that competitive learning is an effective clustering method. As we described in previous sections, competitive learning satisfies a necessary condition for converging to the optimum solution. Hence, the effectiveness of competitive learning for a clustering method must be evaluated to see whether it gives a good approximate solution or not. Since it is reasonable to use an error ratio against the optimum solution for evaluating "goodness" of clustering, we use a 256 gray scale level image for which the optimum solution of clustering can be calculated quickly.

As (11) shows, the number of possible clustering is immense. However, when input vectors are one dimensional, the number of possible clustering $T(m,n)$ is drastically reduced [8] to

$$T(m,n) = \binom{m-1}{n-1}. \tag{29}$$

Furthermore, when the number of levels in the gray scale image equals L, the enumeration corresponds to the case of $m = L$ at (29). Otsu [9] showed that this problem can be solved in $O(nL^2)$ using dynamic programming.

It takes approximately 2 s to get optimum solutions of a 512×512 image when the number of clusters is 8 using a Sun4/200 including I/O overheads. This shows that dynamic programming is suitable to solve the clustering problem in this case. We prepare two images, Lena (picture of girl) and Nasa (remote sensing image). These two images have very different characteristics in histograms. Table I shows optimum solutions of these two images by the Otsu-method.

Now let us compare the optimum solution with experimental results by error ratio r_e defined as

$$r_e\% \equiv \frac{\text{experimental result} - \text{optimum solution}}{\text{optimum solution}} \times 100. \tag{30}$$

We repeated 30 times changing random seeds for each clustering problem. Table II shows the values of parameters for competitive learning and Table III shows average error ratios $\overline{r_e}$ and the probability that r_e is less than 1%. It takes approximately 5 to 20 s to get solutions using a Sun4/200. The computation time depends on the maximum number of

TABLE I
OPTIMUM SOLUTIONS

Image	n	S_W
Lena	2	691.2
	4	162.4
	6	72.78
	8	41.20
Nasa	2	214.0
	4	46.65
	6	21.99
	8	12.65

TABLE II
THE VALUES OF THE PARAMETERS FOR COMPETITIVE LEARNING

n	γ	θ_t	N_{max}
2	0.015	565	5,085
4	0.015	800	44,000
6	0.015	979	114,543
8	0.015	1,131	220,545

TABLE III
ERROR RATIOS AND PROBABILITY OF LOW ERROR

Image	n	$\overline{r_e}$	$r_e < 1\%$
Lena	2	0.10%	100%
	4	0.35%	96.7%
	6	1.13%	60%
	8	0.32%	96.7%
Nasa	2	0.12%	100%
	4	0.30%	96.7%
	6	0.22%	100%
	8	0.35%	100%

repetitions $N_{\max}$. From Table III, competitive learning gives a good approximate solution with high probability, showing that this method is an effective clustering method.

In Table III, the $\overline{r_e}$ of "Lena"($n = 6$) is different from the other values in the table. The phenomenon comes from an existence of a local minimum solution which r_e is only about 2%. Since the S_W of the local minimum solution is close to that of the optimum solution, the probability of converging to the solution cannot be negligible. We think it is not a serious problem, because in more than half the cases, "Lena"($n = 6$) converges to a good approximate solution. In Table II, the value of γ is important to get a good approximate solution in reasonable computation time. However, the values of $N_{\max}$ and θ_t should be changed for this purpose. It is no problem to use the biggest $N_{\max}$ and the biggest θ_t. Experimentally we found the minimum $N_{\max}$ and the minimum θ_t to get a good approximate solution with high probability. These are

$$\theta_t = 400\sqrt{n}, \tag{31}$$

$$N_{\max} = (2n-3)\theta_t(n+7). \tag{32}$$

In (32) $(2n-3)\theta_t$ is the minimum $N_{\max}$ to generate n units (ref. (24)).

We note that the method using dynamic programming is not applicable to color images, because multidimensional data cannot be clustered by thresholds. Therefore, when we apply clustering to color image segmentation, competitive learning becomes meaningful to use. Furthermore, when a gray scale image has intensity level represented by real numbers, it takes $O(nm^2)$ to calculate using dynamic programming and thus the solution by competitive learning becomes more practical.

IV. COLOR IMAGE SEGMENTATION

A. Transformation

We describe a transformation from original vectors to the input vectors which are to be clustered. Tominaga [25] says "Selection of the color space is crucial to the color classification problem, because the arrangement and shape of clusters depend on the color coordinate system selected." The selection of a color space is the selection of this transformation. Let $\boldsymbol{Y}_j (j = 1, \cdots, m)$ be a finite number of original vectors and $(= \mathcal{Y})$, f a transformation function. Then the transformed vectors

$$\boldsymbol{X}_j = f(\boldsymbol{Y}_j)(j = 1, \cdots, m), \tag{33}$$

are input vectors for a clustering procedure. When we use $f(\boldsymbol{Y}_j)$ for $\boldsymbol{X}_j$ in (5), the criterion of clustering to $\mathcal{Y}$ changes according to the function f. We will discuss a transformation to uniform color space (CIE $L*u*v*$) and a linear transformation. Of course, the transformation and the clustering method are independent of each other, so there is no change in the clustering algorithm and its criterion.

The transformation to CIE $L*u*v*$ color coordinate system is a nonlinear transformation described in [24]. In this color coordinate system, the incremental color difference is as follows [24]:

$$\Delta^2\{\text{Color}_1, \text{Color}_2\} = (\Delta L*)^2 + (\Delta u*)^2 + (\Delta v*)^2. \tag{34}$$

Therefore, the clustering criterion of least sum of squares becomes least sum of color differences. It seems that this transformation is the best choice for color clustering. The disadvantage of using this transformation is in taking more time than linear transformations.

The following describes the mathematical features when the transformation is linear and is given by

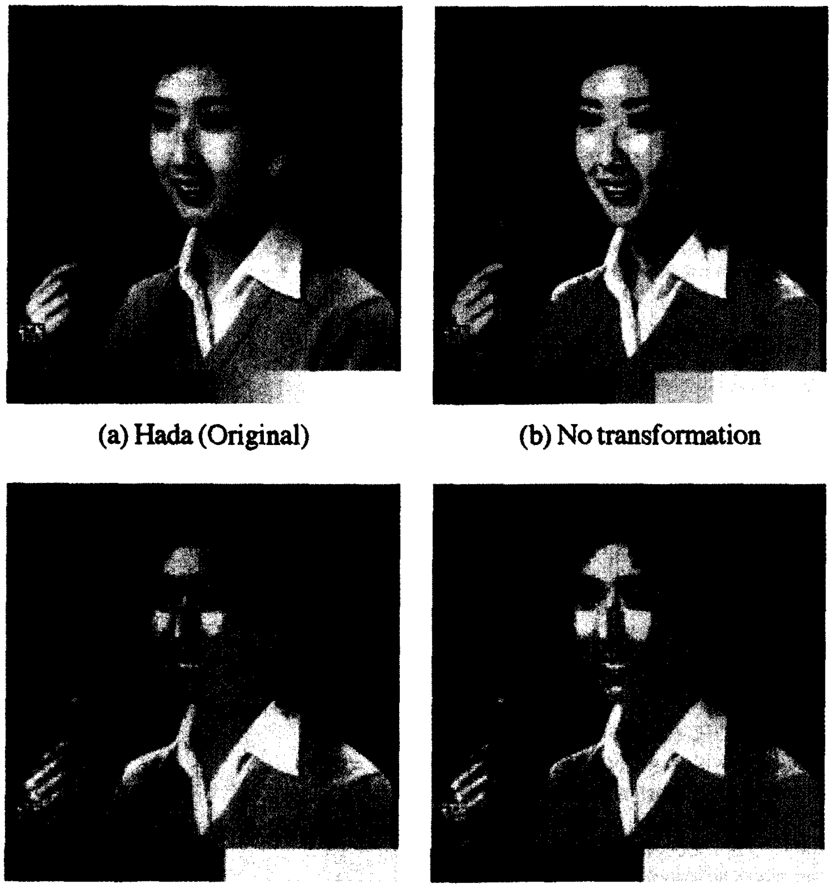
$$\boldsymbol{X}_j = \boldsymbol{T}\boldsymbol{Y}_j \ (j = 1, \cdots, m), \tag{35}$$

where we assume that $\boldsymbol{T}$ is a square real matrix and that $\boldsymbol{T}^{-1}$ can be defined. We define $\boldsymbol{W}_i'$ as

$$\boldsymbol{W}_i' \equiv \boldsymbol{T}^{-1}\boldsymbol{W}_i \ (i = 1, \cdots, n). \tag{36}$$

We have

$$(\boldsymbol{X}_j - \boldsymbol{W}_i) = \boldsymbol{T}(\boldsymbol{Y}_j - \boldsymbol{W}_i'). \tag{37}$$



(a) Hada (Original) (b) No transformation

(c) Whitening transformation (d) L*u*v* transformation

Fig. 4. Segmentation results of "Hada."

We can rewrite (5) and (2) as

$$S_W(\mathcal{C}) = \sum_{i=1}^{n} \sum_{\boldsymbol{Y}_j \in C'_i} p_j (\boldsymbol{Y}_j - \boldsymbol{W}'_i)^t \boldsymbol{A} (\boldsymbol{Y}_j - \boldsymbol{W}'_i) \tag{38}$$

where $C'_i = \{\boldsymbol{Y}_j \mid \boldsymbol{T}Y_j \in C_i\}$ and

$$\boldsymbol{W}'_i = \vec{\mu}'_i \equiv \frac{\sum_{\boldsymbol{Y}_j \in C'_i} p_j \boldsymbol{Y}_j}{\sum_{\boldsymbol{Y}_j \in C'_i} p_j} = \boldsymbol{T}^{-1} \left(\frac{\sum_{\boldsymbol{X}_j \in C_i} p_j \boldsymbol{X}_j}{\sum_{\boldsymbol{X}_j \in C_i} p_j} \right) \tag{39}$$

where $\boldsymbol{A} \equiv \boldsymbol{T}^t \boldsymbol{T}$ and $\vec{\mu}'_i$ is the center of mass in cluster C'_i about $\mathcal{Y}$. Since $\boldsymbol{A}$ is a real symmetric matrix, $\boldsymbol{A}$ can be represented by an orthogonal matrix $\boldsymbol{V}$ and a diagonal matrix $\boldsymbol{K}$ as

$$\boldsymbol{A} \equiv \boldsymbol{T}^t \boldsymbol{T} = \boldsymbol{V}^t \boldsymbol{K} V = (\boldsymbol{K}^{1/2} \boldsymbol{V})^t (\boldsymbol{K}^{1/2} \boldsymbol{V}). \tag{40}$$

Therefore, to minimize $S_W(\mathcal{C})$, it is enough to consider a transformation matrix $\boldsymbol{T}'$ that is presented as $\boldsymbol{K}^{1/2}\boldsymbol{V}$.

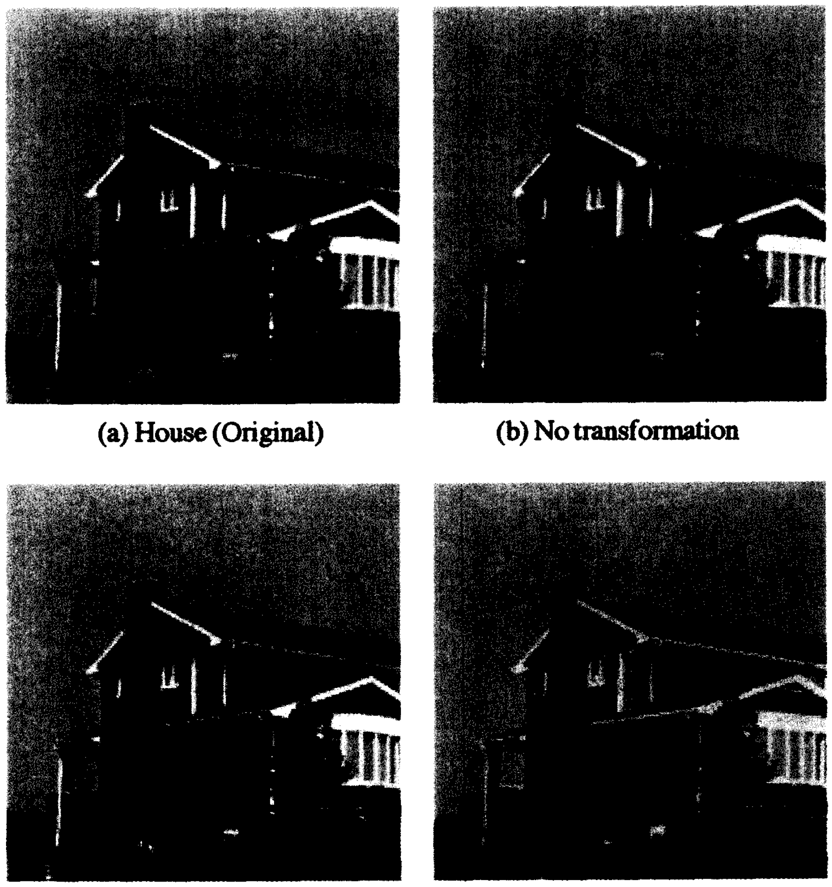
Among the possible transformations, we use the *Whitening transformation* [18] in experiments (Section IV-B). The transformation matrix $\boldsymbol{T}$ is determined by (40), with

$$\boldsymbol{V} = \Phi^t, \boldsymbol{K} = \Lambda^{-1} \tag{41}$$

where Φ is the eigenvector matrix of the original vectors presented by $R_N G_N B_N$ and Λ is their eigenvalue matrix. In the new color space transformed by the matrix, variance along any direction equals 1. We show an effect of this transformation in the following.

B. Experimental Results

Figs. 4 to 7 show results of color image segmentation by clustering. In each figure, picture (a) shows an original image. The names of the images are Hada (Fig. 4), House (Fig. 5), Office (Fig. 6), and Peppers (Fig. 7). They are 512 × 512 images and each pixel of them is presented by $R_N G_N B_N$ (N.T.S.C. receiver primary color coordinate system) with 8 bits each (total 24 bits) of precision. Pictures (b), (c), and (d) are segmentation results. In pictures (b), (c), and (d) each pixel is represented by the color that corresponds to the center of the cluster to which this pixel belongs. When clustering a color space in the experiments, the number of clusters n is 8 and the values of other parameters for competitive learning are the same as listed in Table II. Our method does not decide n. We select 8, because we think 8 colors is enough to represent a characteristic of each image.



(a) House (Original)

(b) No transformation

(c) Whitening transformation

(d) L*u*v* transformation

Fig. 5. Segmentation results of "House."

Pictures (b) are the results of using no transformation, so $R_N G_N B_N$ space is directly clustered based on the least sum of squares. Generally speaking, this color space is divided by differences of intensity rather than hue. For example, skin colors in Fig. 4(b) are divided and the red of the flower is merged into one of the skin colors. The red for the stripes in Fig. 6(b) is merged into yellow. The reason is the following. Table IV shows the contribution ratio of the biggest eigenvalue (λ_1) of the $R_N G_N B_N$ covariance matrix and the eigenvector corresponding to λ_1. As it shows, the contribution ratios are more than 75%, so the clusters are likely allocated along the axis corresponding to λ_1. The direction of the eigenvectors in Table IV are close to an intensity axis, since the intensity axis Y in YIQ coordinate system is defined by [24]

$$Y = 0.299R_N + 0.587G_N + 0.144B_N. \quad (42)$$

Therefore, we use the *Whitening transformation* to divide color space by chromaticity. Pictures (c) are the results of using the *Whitening transformation*. The differences between (b) and (c) are easy to see. In this case, the color spaces are divided by hue as well as intensity. The segmentation results in (c) are better than these in (b), considering "color" clustering. Pictures (d) are the results of using the transformation to the $L{*}u{*}v{*}$ color coordinate system. Since the differences between (c) and (d) are much smaller than between (b) and (c), the Whitening transformation is a good substitute for the transformation to $L{*}u{*}v{*}$.

TABLE IV
CONTRIBUTION OF λ_1 AND ITS EIGENVECTOR

Image	the eigenvector for λ_1			Contribution
	R_N	G_N	B_N	of λ_1(%)
Hada	0.547	0.571	0.612	92.0%
House	0.276	0.605	0.747	91.1%
Office	0.375	0.557	0.741	81.6%
Peppers	0.218	0.855	0.470	76.8%

It is difficult to say which results are good or bad in (c) and (d). The transformation to $L{*}u{*}v{*}$ is reasonable for color clustering because of uniformity in color space. Since the *Whitening transformation* can normalize the distribution, we expect that the transformation cancels reflection caused by the environment. We examine the difference between (c) and (d).

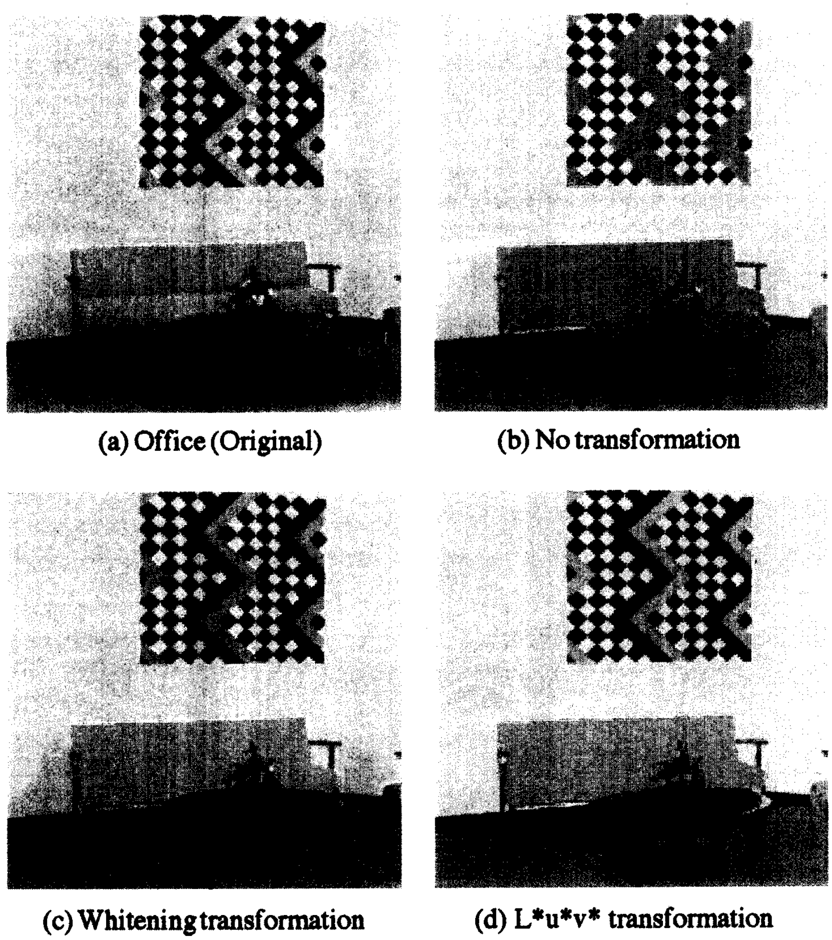


Fig. 6. Segmentation results of "Office."

By the way, we tried to use both the $L{*}u{*}v{*}$ transformation and the *Whitening transformation* together to get a better result. However, we could not find any improvement, because enhancement of the small error in the $L{*}u{*}v{*}$ transformation occurred in the *Whitening transformation.*

In Fig. 4, (c) distinguishes yellow earring from skin color, while (d) does not. As the bars in the bottom of Fig. 4 show, (d) has 4 different clusters in the intensity axis, while (c) has only 3. In Fig. 5, (d) distinguishes roofs and shadows of roofs on walls, while (c) does not. In these cases the *Whitening transformation*, (c), distinguishes hue more than intensity and, the $L{*}u{*}v{*}$ transformation, (d), distinguishes intensity more than hue. This reason for this is the following. These two images have high contrast in intensity, because the contribution of λ_1 (close to intensity axis) is more than 90% in Table IV. The *Whitening transformation* cancels this high contrast in intensity, while the $L{*}u{*}v{*}$ transformation does not.

In Fig. 6, (d) distinguishes yellow stripes and flowers from the chair, while (c) merges these colors. In Fig. 6(d), a part of the highlight on the table is merged into dark color, because black and white are allocated to the same position in the $u{*}v{*}$ plane. The left side of the green bell pepper in front in Fig. 7 is merged into dark shadow in (c), while it is merged into red in (d). In these cases the *Whitening transformation*, (c), distinguishes intensity more than hue and the $L{*}u{*}v{*}$ transformation, (d), distinguishes hue more than intensity. This reason is opposite in the case of Figs. 4 and 5.

We also use just $u{*}v{*}$ coordinate for segmentation. Fig. 8 is the result. In this figure, highlight in the cheek, the collar and the hair are grouped together. This segmentation does not distinguish white, gray, and black. Therefore, information of intensity is also important for color image segmentation, though hue seems to be more important than intensity.

V. Conclusion

We have presented a color image segmentation method which divides color space into clusters using competitive learning. We showed the efficiency of our method from both theoretical and experimental sides. Theoretically, we have shown that competitive learning converges to a local optimum solution of clustering based on the least sum of squares criterion. Experimentally, we showed the efficiency of competitive learning in clustering in gray scale image

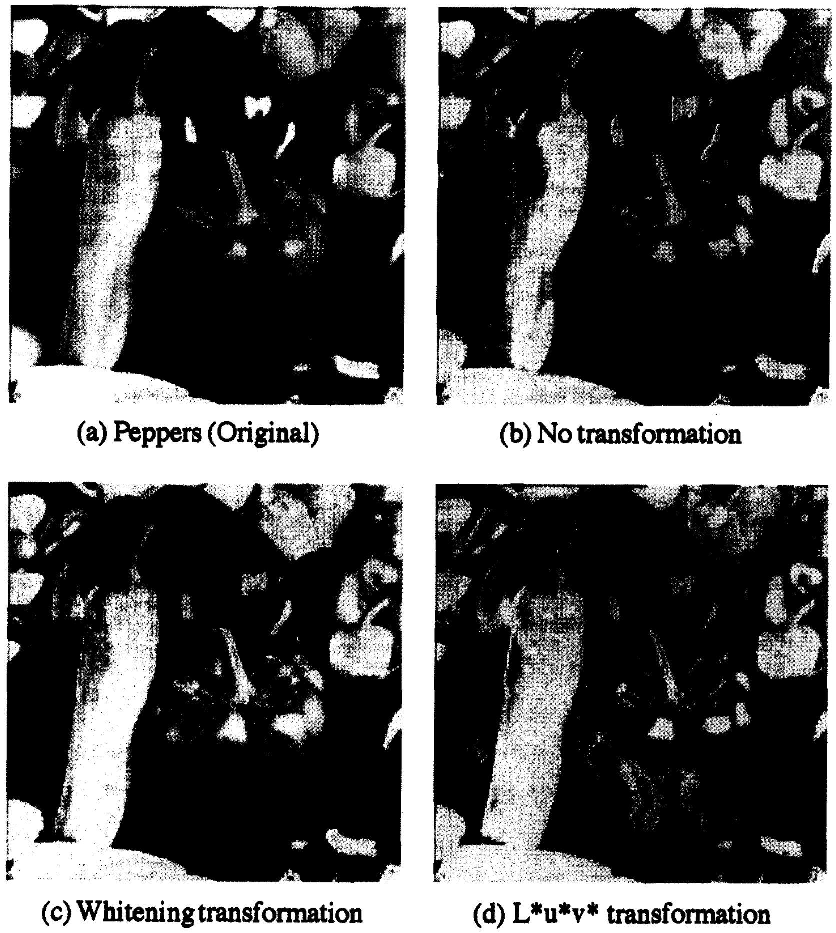


Fig. 7. Segmentation results of "Peppers."

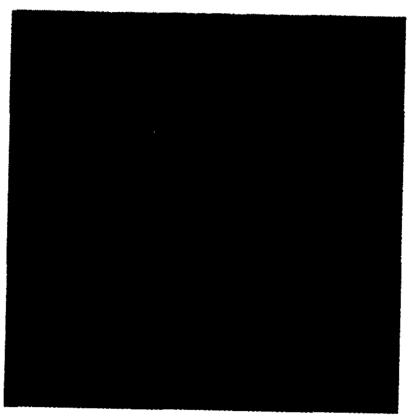


Fig. 8. Segmentation result at $u*v*$ plane.

and showed examples of color image segmentation. We also discussed the effect of transformations that transfer the original vectors to input vectors for clustering.

REFERENCES

[1] E. Riseman and M. Arbib, "Segmentation of static scenes," *Comput. Graphics Image Processing*, vol. 6, pp. 221–276, 1977.

[2] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition*, vol. 13, pp. 3–16, 1981.

[3] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Vision, Graphics Image Processing*, vol. 29, pp. 100–132, 1985.

[4] Y. Fukada, "Spatial clustering procedures for region analysis," *Pattern Recognition*, vol. 12, pp. 395–403, 1980.

[5] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a Tree Traversal Algorithm," *J. Assoc. Comput. Machinery*, vol. 23, pp. 368–388, 1976.

[6] R. Ohlander, K. Price, and D. R. Reddy, "Picture segmentation using a recursive region splitting method," *Comput. Graphics Image Processing*, vol. 8, pp. 313–333, 1978.

[7] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Comput. Vision, Graphics Image Processing*, vol. 13, pp. 222–241, 1980.

[8] H. Spath, *Cluster Analysis Algorithms.* Chichester: Ellis Horwood, 1980, pp. 59–62.

[9] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, pp. 62–66, Jan. 1979.

[10] M. R. Anderberg, *Cluster Analysis for Applications.* New York: Academic, 1973.

[11] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, 1967, pp. 281–297.

[12] G. H. Ball and D. J. Hall, "A clustering technique for summarizing multivariate data," *Behav. Sci.*, vol. 12, pp. 153–155, 1967.

[13] G. B. Coleman and H. C. Andrews, "Image segmentation by clustering," *Proc. IEEE*, vol. 67, pp. 773–785, 1979.

[14] J. Bryant, "On the clustering of multidimensional pictorial data," *Pattern Recognition*, vol. 11, pp. 115–125, 1979.

[15] M. A. Ismail and M. S. Kamel, "Multidimensional data clustering utilizing hybrid search strategies," *Pattern Recognition*, vol. 22, pp. 75–89, 1989.

[16] Q. Zhang and R. D. Boyle, "A new clustering algorithm with multiple runs of iterative procedures," *Pattern Recognition*, vol. 24, pp. 835–848, 1991.

[17] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, 1980.

[18] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* New York: Academic, 1972.

[19] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cognitive Sci.*, vol. 9, pp. 75–112, 1985.

[20] T. Uchiyama, M. Sakai, T. Saito, and T. Nakamura, "Optimum structure learning algorithms for competitive learning neural network," in *Proc. SPIE/SPSE's Electronic Imaging Science & Technology Symp.*, 1991.

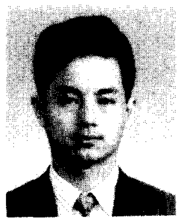
[21] B. Kosko, "Stochastic competitive learning," *IEEE Trans. Neural Networks*, vol. 2, pp. 522–529, 1991.

[22] J. A. Hertz, A. S. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation.* Redwood, CA: Addison-Wesley, 1991, p. 222.

[23] S. Z. Selim and M. A. Ismail, "K-Means-Type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 81–87, Jan. 1984.

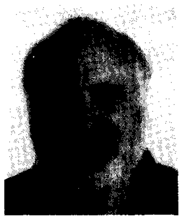
[24] W. K. Pratt, *Digital Image Processing, Second Edition.* New York: Wiley-Interscience, 1991, pp. 62–88.

[25] S. Tominaga, "Color classification of natural color images," *Color Res. Appl.*, vol. 17, pp. 230–239, 1992.



Toshio Uchiyama was born in Japan in 1963. He received the B.S. degree in electrical and electronic engineering and the M.S. degree in physical electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1987 and 1989, respectively.

Since 1989, he has worked for NTT DATA Communications Systems Corporation. His research interests include neural networks, competitive learning, computer vision and image processing. He was a visiting scholar from 1991 to 1993 at the Center for Neural Engineering, University of Southern California, Los Angeles.



Michael A. Arbib (M'66) was born in England in 1940, grew up in Australia (with a B.Sc.(Hons.) from Sydney University) and received the Ph.D. degree in mathematics from MIT in 1963.

After five years at Stanford, he became chairman of the Department of Computer and Information Science at the University of Massachusetts at Amherst in 1970, and remained in the Department until 1986. He then moved to the University of Southern California, where he is now Director of the Center for Neural Engineering, as well as Professor of Computer Science, Neurobiology, Biomedical Engineering, Electrical Engineering, Psychology, and Physiology.