Color Clustering and Learning for Image Segmentation Based on Neural Networks

Guo Dong, *Member, IEEE,* and Ming Xie, *Member, IEEE*

***Abstract*—An image segmentation system is proposed for the segmentation of color image based on neural networks. In order to measure the color difference properly, image colors are represented in a modified $L^*u^*v^*$ color space. The segmentation system comprises *unsupervised* segmentation and *supervised* segmentation. The unsupervised segmentation is achieved by a two-level approach, i.e., color reduction and color clustering. In color reduction, image colors are projected into a small set of prototypes using self-organizing map (SOM) learning. In color clustering, simulated annealing (SA) seeks the optimal clusters from SOM prototypes. This two-level approach takes the advantages of SOM and SA, which can achieve the near-optimal segmentation with a low computational cost. The supervised segmentation involves color learning and pixel classification. In color learning, color prototype is defined to represent a spherical region in color space. A procedure of hierarchical prototype learning (HPL) is used to generate the different sizes of color prototypes from the sample of object colors. These color prototypes provide a good estimate for object colors. The image pixels are classified by the matching of color prototypes. The experimental results show that the system has the desired ability for the segmentation of color image in a variety of vision tasks.**

***Index Terms*—Color clustering, color leaning, color reduction, color space, reduced Coulomb energy (RCE), self-organizing map (SOM), simulated annealing (SA), supervised segmentation, unsupervised segmentation.**

I. Introduction

IMAGE segmentation is the first important process in numerous applications of computer vision. It partitions the image into different meaningful regions with homogeneous characteristics using discontinuities or similarities of image components, the subsequent processes rely heavily on its performance. In most cases, the segmentation of color image demonstrates to be more useful than the segmentation of monochrome image, because color image expresses much more image features than monochrome image. In fact, each pixel is characterized by a great number of combination of R, G, B chromatic components—2^{24}. However, more complicated segmentation techniques are required to deal with rich chromatic information in the segmentation of color images.

According to the usage of prior knowledge of the image, color image can be segmented in a *unsupervised* or *supervised* way. The former attempts to construct the "natural grouping" of the image without using any prior knowledge. The latter, however, separates the image based on the sample of object colors. The unsupervised segmentation is widely used in the applications where the image features are unknown, such as nature scene understanding, satellite image analysis etc. The supervised segmentation is commonly used in the applications where the sample of object colors can be acquired in advance, e.g., object tracking, face/gesture recognition, and image retrieval etc. A complete system of image segmentation should include both unsupervised and supervised segmentation. It should be a self-supporting system that requires the unsupervised segmentation to work cooperatively with the supervised segmentation.

The segmentation of color image requires the computational cost considerably higher than that needed for monochrome image, but it is no longer a major problem with the increasing speed of computation and decreasing cost of color sensors. In fact, there has been a remarkable growth of techniques for the segmentation of color images in this decade. A variety of segmentation techniques have been proposed in the literature. However, most techniques are kind of "dimensional extension" directly inherited from the segmentation of monochrome image [1], [2].

A. Unsupervised Segmentation

The spatial compactness and color homogeneity are two desirable properties in unsupervised segmentation, which lead to *image-domain* and *feature-space* based segmentation techniques. According to the strategy of spatial grouping, image-domain techniques include *split-and-merge*, *region growing* and *edge detection* techniques. There have been extensive studies on them in the literature. In [3], the Markov random field (MRF) is defined in the quad-tree structure to represent the continuity of color regions in the process of split-and-merge. In [4], splitting and merging phases are operated by the *watershed transform* and self-organizing map (SOM), respectively. Zhu developed a segmentation algorithm named as "Region Competition" in [5]. It combines the global optimization methods (snakes/balloons and region growing) to guarantee the convergence of global optima. Manjunath [6] defined the J-image using local windows in a quantized class-map. The high and low values in J-images correspond to possible boundaries and centers of the regions. The minimum vector dispersion (MVD) operator is proposed to reduce the color vector to a scalar value in [7]. It is a bias free operator for step edges which produces a strong response for true ramp edges. A circular compass operator is proposed to detect color edges in [8]. The orientation of 'needle' with the maximum difference indicates the edge direction, and its magnitude yields a measure of edge strength.

Manuscript received June 24, 2003; revised December 17, 2004. The work of G. Dong was supported by the Nanyang Technological University under a Research Fellowship.

G. Dong is with DSO National Laboratories, Singapore 118230, Singapore (e-mail: guodong@ieee.org).

M. Xie is with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: mmxie@ntu.edu.sg).

Digital Object Identifier 10.1109/TNN.2005.849822

In feature-space based techniques, image segmentation is accomplished by exploiting the homogeneous regions in feature space. The common techniques include *histogram thresholding* and *color clustering*. The histogram thresholding is a technique that seeks the peaks or valleys in 3 color histograms or a three-dimensional (3-D) histogram. The HSV histograms are used for the segmentation of color image in [9]. The achromatic regions are determined by the saturation values, and the remaining chromatic regions are segmented by thresholding the peaks of hue histogram. A 3-D color histogram is built by $L^*u^*v^*$ color components in [10]. The valleys of color histogram are identified by the *watershed* algorithm.

The *nonparametric clustering* is a promising solution in color clustering. The standard techniques can be categorized as *hierarchical* or *partitional* clustering [11]. In hierarchical clustering, only local neighbors involve the cluster merging/splitting by the form of dendrograms. The global knowledge of clusters is not incorporated to the procedure of clustering. The partitional clustering, on the other hand, is an iterative procedure that directly decomposes the data set into a number of disjoint clusters by minimizing the criterion function (e.g., sum-of-squared-error). The k-means and ISODATA are well-known techniques of partitional clustering. However, they suffer the problems of local optima, clustering reproducibility and initialization sensitivity. The k-mean and ISODATA clustering require the number of clusters to be known *a priori*. In order to determine the optimal number of clusters, Turi [12] proposed a validity measure using the ratio of intra-cluster and inter-cluster measures incorporated with a Gaussain multiplier. The optimal number of clusters is found by minimizing the validity measure.

Some new techniques have been proposed for color clustering in the literature. Comaniciu employed the *mean shift* analysis for the exact estimation of clustering kernel in [13]. The spheres with the predefined size are used to search the centers of color clusters in color space. It has shown the good performance on segmenting the images with strong variations of density. Shi [14] treated the segmentation as a graph partitioning problem. A *weighted undirected graph* is constructed by taking the pixels as the nodes of graph. The edges are defined between each pair of nodes, whose weights are the product of color similarity and spatial proximity terms. The graph is bipartitioned by optimizing the criterion of *normalized cut* based on the computation of eigenvalues.

B. Supervised Segmentation

In supervised segmentation, the pixel classifier is trained for the best partition of color space using the sample of object colors. The image is segmented by assigning the pixel to one of the predefined classes. The common techniques of supervised segmentation are evaluated in [15], including *maximum likelihood*, *decision tree*, *nearest neighbor* and *neural networks*. The supervised segmentation is employed for the segmentation of video shots in [16]. The segmentation of image frames is hierarchized by three classifiers, i.e., *k nearest neighbor*, *näive bayes*, and *support vector machine*. In [17], image segmentation is performed by a procedure of supervised pixel classification. The rule of *minimum distance decision* is used to assign each pixel to a specific class in a color texture space.

C. Artificial Neural Networks

Artificial Neural Networks offer many important properties to the segmentation of color image, e.g., *high degree of parallelism*, *nonlinear mapping*, *adaptivity*, and *error tolerance* etc. Different types of neural networks have been proposed for the segmentation of color image. In unsupervised segmentation, Uchiyama [18] employed the competitive learning (CL) for the online color clustering based on the least sum of squares criterion. The CL procedure is able to converge to a local optimum solution for color clustering. Scheunders compared the performance of CL clustering with other clustering algorithms for color quantization in [19], i.e., CMA, GCMA, and HCL. The evaluations show that HCL and GCMA are insensitive to the initial conditions. The GCMA produces the most optimal results with a high computational cost, but HCL can reach the near-optimal results with a low computational cost. A two-stage clustering approach is proposed for the fast clustering in [20]. The CL is used to identifying the local density centers of the clustering data. The iterative Gravitation Neural Network is employed to agglomerate the resulting codewords in a parallel fashion.

SOM has been used for the segmentation of color image in the literature. In [21], SOM is trained to generate the primitive clustering results based on the set of five-dimensional (5-D) vectors (R, G, B, and x, y coordinates). The image is segmented by merging the scattered blocks and eliminating the isolated pixels. The SOM is used to capture the dominant colors of the image in [22]. These dominant colors are further merged to form final color clusters. In [23], color quantization was implemented by a one-dimensional (1-D) SOM. The acceptable quantization is achieved by dynamically expanding or contracting the 1-D SOM. A frequency-sensitive SOM (FS-SOM) is proposed for the color quantization in [24]. The frequency sensitive learning scheme is able to improve the quality of color quantization by localizing the learning rate to each neuron.

The SOM clustering is fully studied by the two stages in [25], i.e., using SOM to produce the prototypes that are clustered in the second stage. Some clustering algorithms are employed for the prototype clustering. It is found that the two-level approach demonstrates better clustering results and lower computational cost than the direct clustering. The two-stage SOM clustering has been used for the segmentation of color image in [26]. The main colors of the image is obtained by SOM mapping. An optimal clustering process is used to determine the number of clusters and group the SOM prototypes. A hardware accelerator of SOM is implemented in [27]. It allows the efficient usage of silicon areas for the different sizes of SOM maps.

Simulated annealing (SA) has been used to gain the best overall segmentation of color image in [28]–[31]. The SA is proposed for segmenting SAR images in [28]. It provides a good solution for the common problem of SAR image segmentation [28], i.e., the poor placement of region edges. The MRF is introduced to model spatial and color contexts among image pixels in [29], [30]. The SA is used to seek the MAP of MRF parameters. In [29], an invariant transform is proposed to remove highlight and shading effects using Dichromatic Reflection Model. Realizing SOM clustering is susceptible

to being trapped to the local optima, SA is used to control the updates of representative patterns during SOM learning in [30]. The segmentation is speeded up by performing the SA on image pyramids in [31].

In supervised segmentation, Campadelli [32] proposed two segmentation algorithms based on *Hopfield network*. The segmentation of color image is formulated as the minimization of a suitable energy functions. Littmann [33] used the local linear maps (LLM) networks to segment a hand gesture from the background scene. The LLM is trained by the sample of skin colors. It yields a probability for the pixels belonging to the region of hand gesture.

D. Description of Problem

The segmentation of color image has been proved to be difficult because it involves a vast amount of data processing. Although great efforts have been devoted to it, some issues are still not fully addressed. In color representation, *perception uniform* guarantees the proper measure of color differences, which is often neglected in many segmentation methods. A number of color spaces have been used in the segmentation of color image. However, colors are highly correlated in the *nonuniform* color space, it is inaccurate to estimate the color differences by the distance measure. To ensure a proper measure of color differences, image colors must be represented in a *uniform color space*.

In unsupervised segmentation, *color reduction* is indispensable to the segmentation of a large color image. It produces a preserving projection of original color data with the minimum distortion, and replaces them with 'representative prototypes' [34]. The computational cost is greatly reduced when the clustering is further performed on the prototypes. *Color clustering* is a promising solution for unsupervised segmentation. It seeks the color clusters by minimizing or maximizing the criterion function (e.g., minimum variance criterion) to the utmost extent. As the structure of color data is complicated, multiple local optima often exist in color data. Many clustering algorithms, like k-mean clustering, are affected by the starting configurations and often get stuck at local optima. Hence, *global optimization* is the key issue to get the optimal clustering results.

In supervised segmentation, *color learning* is crucial to build up an accurate classifier for the segmentation of the object of interest. Due to the variations of lighting, color distributions of interested objects may be complicated and irregular in color space. The simple techniques, such as color thresholding, are not accurate enough to differentiate the object colors from the background. In supervised segmentation, it is important that color learning is able to build up an accurate color model for the object of interest.

E. System Structure

A complete segmentation system is developed for both unsupervised and supervised segmentation of color image. Fig. 1 shows the structure of the system. The image colors are represented in a Modified $L^*u^*v^*$ color space for unsupervised and supervised segmentation. The system consists of four computational modules: color reduction and color clustering in the unsupervised segmentation; color learning and pixel classification in the supervised segmentation. These computational modules provide a versatile tool for the segmentation of color images. The unsupervised segmentation plays a supporting role in the supervised segmentation. In color learning, the sample of object colors can be obtained by the unsupervised segmentation of the sample image.

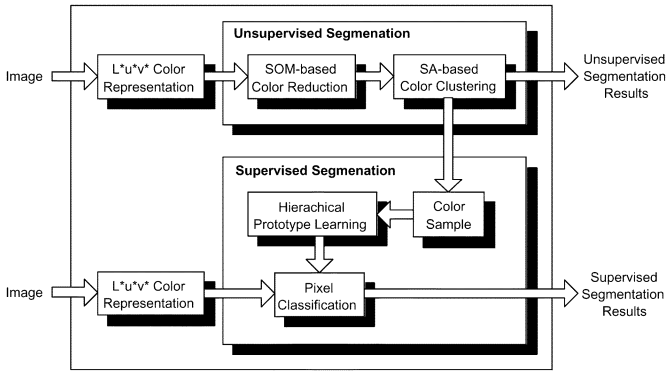

Fig. 1. Image segmentation system based on neural networks.

The paper is organized as follows. In Section II, an appropriate color space is introduced to represent the image colors. In Section III, color reduction is performed by SOM learning based on its nonlinear and topology-preserving properties. The SA is employed to seek the optimal clusters from SOM prototypes. A new procedure of supervised learning is discussed in Section IV. The proposed segmentation system has been used in many vision tasks, Section V demonstrates the experimental results and some applications. The paper is concluded in Section VI.

II. Color Representation

A number of color spaces have been used in the segmentation of color image, such as RGB, CIE XYZ, CIE $L^*u^*v^*$, CIE $L^*a^*b^*$, HSL, YUV and YIQ etc. A color space is uniform, if the equal distance in the color space corresponds to equal perceived color differences. Many color spaces are nonuniform. For example, RGB color space is far from exhibiting the perceptual uniformity, it does not model the way that human perceive colors. In HSL color space, different computations are involved around 60°, which introduce the visible discontinuities in color space. Riemersma [35] studied four color spaces that score well in perceptual uniformity, i.e., Nonlinear $R'G'B'$, YUV, CIE $L^*u^*v^*$, and modified CIE $L^*u^*v^*$. It is concluded that the Modified CIE $L^*u^*v^*$ performs better than other color spaces. In this paper, image colors are represented in a Modified $L^*u^*v^*$ color space. The CIE $L^*u^*v^*$ color space is defined from the CIE standard color model XYZ. The conversion from RGB to Modified CIE $L^*u^*v^*$ includes the following three steps.

A. RGB to CIE XYZ

In accordance with the ITU-R recommendation BT.709, CIE XYZ can be transformed from RGB by a 3×3 matrix transform

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412\,453 & 0.357\,580 & 0.180\,423 \\ 0.212\,671 & 0.715\,160 & 0.072\,169 \\ 0.019\,334 & 0.119\,193 & 0.950\,227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (1)$$

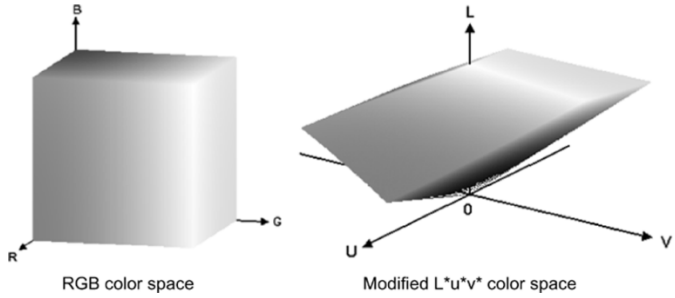

Fig. 2. Color distributions in RGB and modified $L^*u^*v^*$ color spaces. In RGB color space, R, G, and B range from 0 to 255. In modified $L^*u^*v^*$ color space, $L^* \in [0, 159.69]$, $u^* \in [-127.86, 242.09]$, and $v^* \in [-179.49, 169.13]$.

B. CIE XYZ to CIE $L^*u^*v^*$

The computation of CIE $L^*u^*v^*$ involves intermediate u' and v' quantities, where the prime denotes the successor to obsolete 1960 CIE u and v system

$$u' = \frac{4X}{X + 15Y + 3Z} \quad v' = \frac{9Y}{X + 15Y + 3Z}. \tag{2}$$

We compute u'_n and v'_n for the reference white X_n, Y_n and Z_n. The D65 white point of Rec.709 is used by default, which has the coordinates fixed as $(x_n, y_n) = (0.3127, 0.3290)$[1]. Then, $(u'_n, v'_n) = (0.1978, 0.4683)$. The luminant component L^* is computed by [36]

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16 & \text{if } \frac{Y}{Y_n} > 0.008\,856 \\ \frac{903.3Y}{Y_n} & \text{otherwise.} \end{cases} \tag{3}$$

The chromatic components u^* and v^* are given as follows:

$$u^* = 13L^*(u' - u'_n) \quad v^* = 13L^*(v' - v'_n). \tag{4}$$

C. CIE $L^*u^*v^*$ to Modified $L^*u^*v^*$

The standard CIE $L^*u^*v^*$ space is designed strictly in laboratory conditions with adapted observers. In [37] and [38], Hunt and Nemcsics studied the perception in a complex environment. It is concluded that the brightness L is proportional to $\sqrt{Y}$ rather than $\sqrt[3]{Y}$ in CIE $L^*u^*v^*$ formula for the complex viewing environment. Matkovic recommended the formulas of CIE $L^*u^*v^*$ color space as follows [39]:

$$L^* = 10\sqrt{Y} \quad u^* = 13L^*(u' - u'_n) \quad v^* = 13L^*(v' - v'_n). \tag{5}$$

The above L^*, u^*, v^* are used to represent the image colors in the segmentation system. Fig. 2 shows the comparison of color distribution in RGB and Modified $L^*u^*v^*$ color space. In Modified $L^*u^*v^*$ color space, L^* is luminant component, u^* and v^* are color components, u^* axis varies from green to red, and v^* axis changes from blue to yellow.

III. Unsupervised Segmentation

The image colors appear to be a massive and noisy nature in the color space. The common clustering approaches are often prone to being trapped into local optima. The SA color clustering is a good solution of seeking the optimal color clusters from a global scope. However, it is far from the practical usage due to the high computational cost. We use the SA clustering for the segmentation of color images by a two-level color clustering, i.e., SOM-based color reduction and SA-based color clustering. The color reduction is the first step in the two-level color clustering. It is especially useful for the segmentation of a large color image. As stated in [25], color reduction can lessen the computational cost, and diminish the sensitivity of clustering to the noise data.

A. Color Reduction

The color reduction maps a given set of 3-D color points into a surrogate lower dimensional space. Let $\Re^3$ be a 3-D color space, it carries out a projection of color points from color space $\Re^3$ into a lower dimensional space $\Re^d$ by the transformation T

$$\mathbf{X} \in \Re^3 \stackrel{T}{\Longrightarrow} \mathbf{Y} \in \Re^d, \quad d < 3. \tag{6}$$

Some linear or nonlinear techniques of dimensionality reduction can be used for the color reduction, such as PCA and ICA. In comparison with these techniques, SOM has the advantages of nonlinear projection, topology preserving and prominent visualization, which makes it particularly useful for the color reduction [40]. Given a set of color points $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x}_n\}$ in modified $L^*u^*v^*$ color space $\Re^3_{luv}$, it is trained to map $\mathbf{X}$ onto a two-dimensional (2-D) array of nodes $\mathbf{M}$ such that color points neighboring in $\Re^3_{luv}$ are projected to nearby nodes in $\mathbf{M}$.

The SOM is structured as a two-layer neural network with a rectangular topology as shown in Fig. 3(a). Three inputs (L^*, u^* and v^*) are fully connected to the neurons on a 2-D plane. Each neuron i is a cell containing a template against which inputs are matched. The template is the weight values to the neuron i, which is represented by $\mathbf{w}_i = [w_{i1}, w_{i2}, w_{i3}]^T$. The SOM training has the following procedure.

1) Initialization: Define the SOM map. The weight vector $\mathbf{w}_i(0)$ of the ith neuron is randomly initialized. By default, we set the size of SOM map to be 16×16. The topology type is rectangular. The neighborhood type is Gaussian. The SOM training is successively performed by two phases. The neighborhood radius $r = 16$, 5, and learning rate $\alpha = 0.05$, 0.02. The weight vectors of the map are ordered in the first phase, and fine-tuned in the second phase.

2) Input: The image colors are reiteratively used to train the network for few times. During the training, each color point $\mathbf{x}$ is cyclically chosen from the data set, and presented to all neurons on the map simultaneously.

3) Competitive Process: At time t, color point $\mathbf{x}(t) = [l(t), u(t), v(t)]^T$ is presented to the network. The 'winning neuron' c is computed with the shortest distance between color point and weight vectors by

$$\|\mathbf{x}(t) - \mathbf{w}_c(t)\| = \min_i \{\|\mathbf{x}(t) - \mathbf{w}_i(t)\|\} \tag{7}$$

where

$$c = \arg\min_i \{\|\mathbf{x}(t) - \mathbf{w}_i(t)\|\}.$$

[1]See Question 16 in [36].

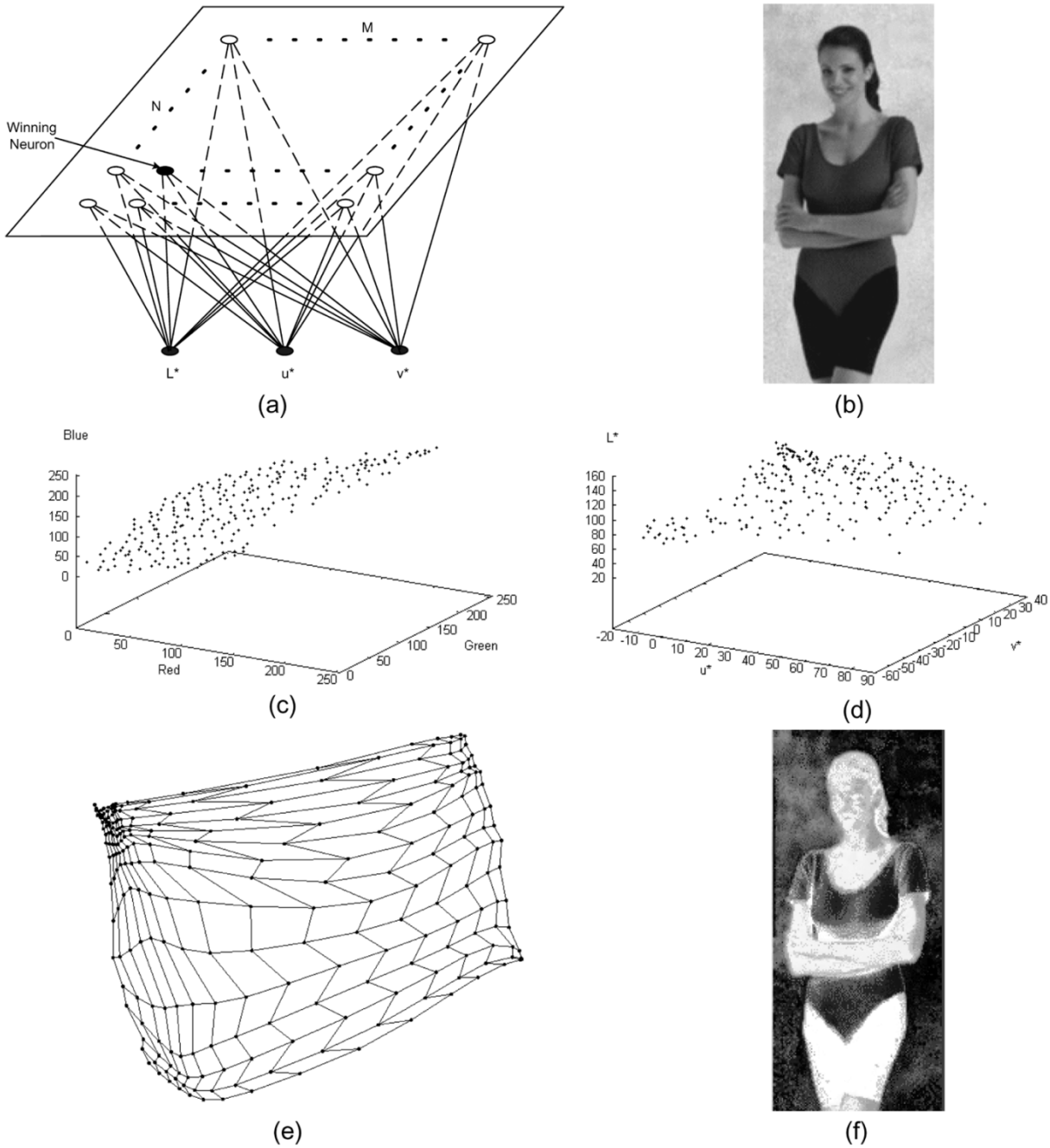

Fig. 3. Color reduction by SOM. (a) SOM used for color reduction. (b) The original color image. (c) Distribution of color points in RGB color space. (d) Distribution of color points in modified $L^*u^*v^*$ color space. (e) Sammon mapping of 16×16 weight vectors after SOM training. (f) Result of color reduction is displayed as a monochrome image. The gray level stands for one of 256 neurons.

4) Cooperative Process: The topological neighbors of 'winning neuron' c are determined by Gaussian function centered at neuron c with the effective scope of $R_c(t)$.

5) Adaptive Process: The weights of "winning neuron" c and its neighbor neurons are updated within the neighborhood

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w_i}(t) + \alpha(t)h_{ci}(t)[\mathbf{x}(t) - \mathbf{w_i}(t)] & \text{if } i \in R_c(t) \\ \mathbf{w}_i(t) & \text{otherwise} \end{cases} \quad (8)$$

where, $\alpha(t)$ is the learning factor, and $h_{ci}(t)$ is the neighborhood function centered around the winning neuron c.

6) Iteration: The next color point is presented to the network at time $t+1$. The learning rate α is linearly decreased to $\alpha(t+1) = \alpha(0)(1.0 - t/T)$. The neighborhood radius R is linearly reduced to $R(t+1) = R(0)(2.0 - t/T)$, where T is the number of color points for training. The new 'winning' neuron is chosen by repeating the procedure from step 2 until all iterations have been made ($t = T$, $\alpha = 0$, $R = 1$).

The trained feature map is a topologically ordered map where the spatial location of a neuron in the lattice corresponds to a particular subset of color points in color space. The Sammon mapping is used to obtain the spatial relationships of neurons. Fig. 3 shows an example of SOM-based color reduction. The RGB colors of the image are shown in Fig. 3(c), and converted to

$L^*u^*v^*$ colors in Fig. 3(d). The SOM map is trained by the color data in Fig. 3(d). The Sammon mapping of weight vectors is shown as a griding map in Fig. 3(e). In color reduction, all color points are presented to the SOM map, and the best matching neuron is searched from the map. In this way, each color point is characterized by one neuron. Fig. 3(f) shows the results of color reduction.

B. Color Clustering

The color reduction transforms $\mathbf{X}$ into $\mathbf{M}$ based on SOM learning, where M is a set of 2-D vectors $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{MN}\}$. Given the number of color clusters K, color clustering attempts to organize the data set $\mathbf{M}$ into a set of clusters $\mathbf{C} = \{\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_K}\}$, such that the vectors in a cluster $\mathbf{c_i}$ are 'more similar' than the vectors belonging to other clusters. The SA mimics the principle of annealing in the physical domain. The optimal solution is obtained by consisting in randomly perturbing the system, and gradually decreasing the randomness to a low final level. It provides a good solution for the color clustering.

Let the set of cluster centers be $\mathbf{Z} = \{\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_K}\}$. The criterion of sum-of-squared-error J is defined as the energy function of SA to quantify the homogeneity of clusters

$$J(\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_K}) = \sum_{i=1}^{K} \sum_{\mathbf{m_j} \in \mathbf{c_i}} \|\mathbf{m_j} - \mathbf{z_i}\|. \tag{9}$$

The procedure of SA clustering is to search the appropriate cluster centers $\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_K}$ that minimizes the energy function J. The SA clustering starts with randomized states throughout the data set $\mathbf{M}$ from a high temperature $T_{\max}$. The data set $\mathbf{M}$ are initially assigned to K clusters randomly, and cluster centers are computed by

$$\mathbf{z_i} = \frac{1}{n_i} \sum_{\mathbf{m_j} \in c_i} \mathbf{m_j}. \tag{10}$$

Next, a point is randomly redistributed from the cluster $\mathbf{c_i}$ to the cluster $\mathbf{c_j}$. The energy is changed from J to J'. If the new candidate state has a lower energy, the state of redistribution is accepted. Otherwise, the state of redistribution is accepted if the probability of $e^{-((J'-J)/T)}$ is higher than a random probability. The algorithm continues redistributing the point to different clusters randomly till each point is redistributed several times. Then, T is lowered, the point redistribution is repeated, and so forth. When the T is sufficient low, the clustering is in a low-energy state, which means that the J is minimized. Given the number of point redistribution N_T, the process of SA clustering is summarized as follows:

1) $T = T_{\max}$, N_T, generate the initial clusters with energy J by randomly distributing the points to K clusters;
2) *while* $T > T_{\min}$;
3) *for* $j = 1$ *to* N_T, $j = j + 1$;
4) select the point randomly from the cluster c_i, redistributed it to the cluster c_j, compute the energy J';
5) if $J' < J$, accept the new state of clustering;
6) else if $e^{-((J'-J)/T)} > \mathrm{Rand}[0, 1)$, accept new state of clustering;
7) *otherwise*, reject the new state of clustering;
8) end *for*;
9) decrement T;
10) end *while*.

The asymptotic convergence of annealing is often performed exponentially to guarantee the small changes of temperature. In order to reduce the computational cost, we use a simple decrement function as follows:

$$T_t = \alpha T_{t-1} \tag{11}$$

where α is annealing factor, $\alpha = 0.8$–0.99.

Fig. 4 shows an example of SA clustering. The set of 2-D points in Fig. 4(a) is obtained from the Sammon mapping in Fig. 3(e). The 2-D points are clustered to four classes in SA-based color clustering. The descent of energy function is shown in Fig. 4(b). By combining the results of SOM-based color reduction, we obtain the final color clustering results. Fig. 4(c) shows the clustering results in the modified $L^*u^*v^*$ color space, and Fig. 4(d) shows the segmentation results.

IV. Supervised Segmentation

The supervised segmentation includes color sample learning and pixel classification. The former is used to build up the pixel classifier. The latter segments the image based on the pixel classifier. The color distribution of the object is often complicated and nonuniform in the color space. An example of hand gesture is shown in Fig. 5(a). We use the SOM-SA color clustering to segment human hand from the image background. The segmentation results are shown in Fig. 5(b) and (c). The distribution of hand colors is shown in Fig. 5(d). It shows a very irregular appearance in color space. In supervised segmentation, it is crucial to build up an accurate and efficient color model for the object based on the sample learning. In order to represent the complex distribution of object colors, we proposed a new learning procedure by extending the learning principle of reduced Coulomb energy (RCE) neural network.

A. RCE Learning

RCE neural network is a supervised pattern classifier used for the estimation of feature region [41]. As stated in [42], it provides a way of region adjustment that is intermediate between Parzen-window and K-nearest-neighbor. During the network training, the size of the hyperspherical window is adjusted in reference to the nearest point of a different category in feature space. The feature region of each category is enclosed by generated hyperspherical prototypes.

It is not advisable to directly use RCE neural network for the supervised segmentation. The main drawback of RCE learning is the requirement of a complete sample set for all classes. In order to segment the object of interest from the image background, it requires the samples of both the object and the image background. However, it is impractical to obtain the sample of the image background when the image background unpredictably changes over the time. Another drawback is that all prototypes are generated with the fixed size in one class, which are not accurate and efficient for region estimation. This drawback can be revealed in Fig. 5(e) and (f). The region of hand colors is estimated by two different sizes of color prototypes. The color prototypes are generated by the following procedure,

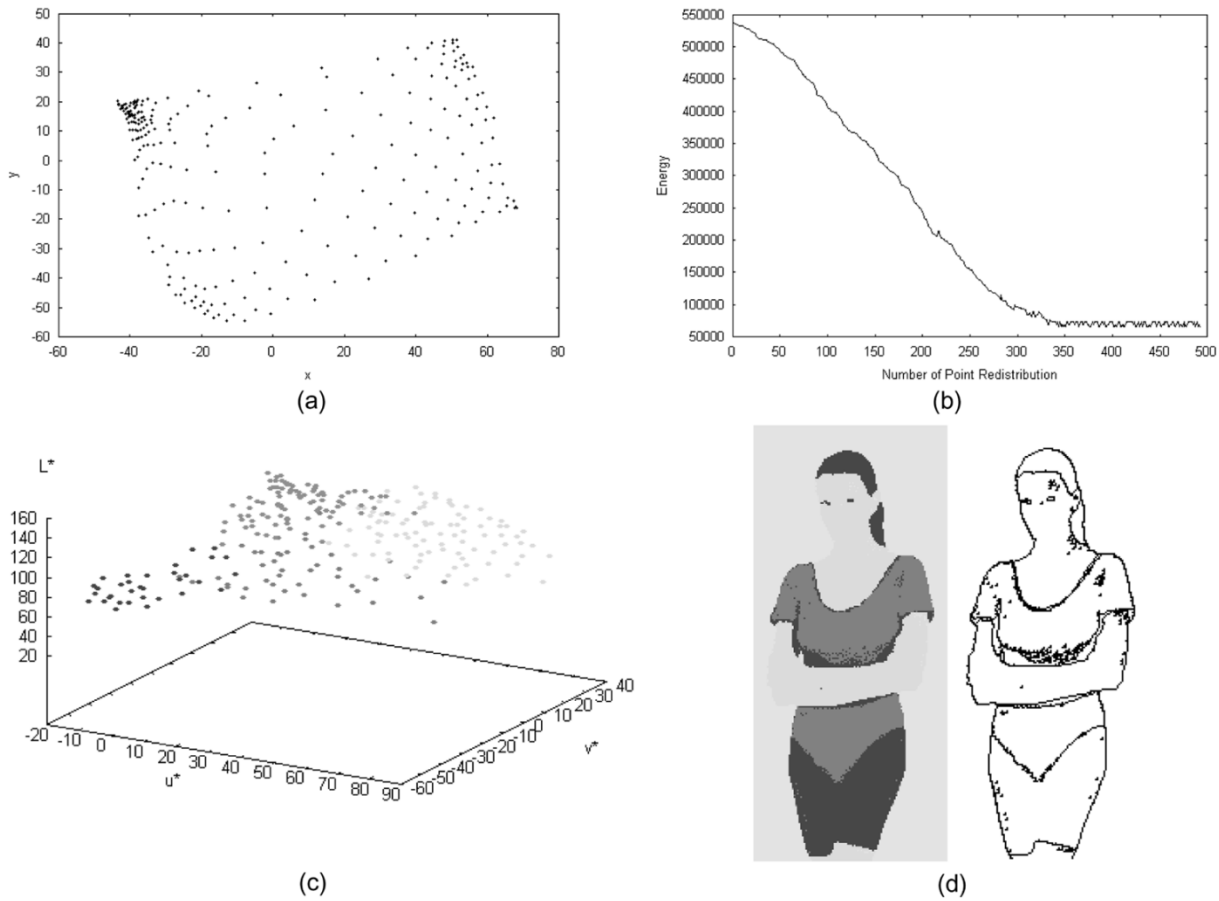

Fig. 4. Color clustering by SA. $T_{\max} = 100$, $T_{\min} = 0.001$, $\alpha = 0.98$, and $N_T = 1000$. (a) The 2-D points in SOM sammon mapping $\mathbf{Y}$. (b) Energy descent during SA clustering. (c) Clustering result in the modified $L^*u^*v^*$ color space. (d) Segmentation result by SOM-SA color clustering.

1) Architecture: The RCE neural network includes 3 layers. In the input layer, three neurons receive the $L^*u^*v^*$ color components from the sample. All color prototypes are stored in the prototype layer. Each color prototype is fully connected with three neurons in the input layer. To learn the sample of hand colors, one neuron is defined to represent the class of human hand in output layer. All the prototypes of hand colors are linked to it.

2) Learning: The RCE neural network is trained by the sample of hand colors in Fig. 5(d). Let the size of spherical prototypes be $R = 5.0$, 20.0, respectively. Fig. 5(e) and (f) shows the color prototypes generated from RCE learning.

3) Segmentation: Given the test image, the supervised segmentation is performed by classifying each pixel as human hand and image background. The pixel is classified as human hand if it falls into any of color prototypes. Otherwise, it is regarded as the image background.

In Fig. 5(e), the region of hand colors is well enclosed by the color prototypes with the low radius $(R = 5.0)$, but a great number of color prototypes cause a heavy computation load in the pixel classification. Fig. 5(f) shows the region of hand colors is over-estimated by the color prototypes with the high radius $(R = 5.0)$. The image is over-segmented in this case.

B. Hierarchical Prototype Learning

The color region of human hand can not be well estimated by the fixed size of color prototypes in RCE training. This is due to the nonuniform distribution of object colors. In some regions, a small size of prototype is appropriate, while elsewhere a large size of prototype is more suitable. The proper way of region estimation is to estimate the region by the different sizes of color prototypes. We proposed a new learning procedure, namely hierarchical prototype learning (HPL), to generate the multiscale color prototypes from the training sample.

1) Color Prototype: Given the training sample $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$, $\mathbf{x_i} \in [l_i, u_i, v_i]^T$. Assume that a color prototype $\mathbf{p}$ is centered at $\mathbf{c}$ with the radius r. In color space, it represents a spherical region centered at $\mathbf{c}$ with the radius r. Given a color point $\mathbf{x_i}$, it is committed to $\mathbf{p}$ if it falls into the spherical region of $\mathbf{p}$. Define $d_{(x_i,p)}$ as Euclidean distance between $\mathbf{x_i}$ and $\mathbf{p}$

$$d_{(x_i,p)} = \left| \sum_{j=1}^{3} (c_j - x_{ij})^2 \right|^{1/2}. \tag{12}$$

Then, $\mathbf{x}$ is committed to $\mathbf{p}$ only if $d_{(x_i,p)} \leq r$.

2) Density of Prototype: Let N be the number of color points that are committed to prototype $\mathbf{p}$ in the training sample. The density of prototype $\mathbf{p}$ is computed by

$$D_p = \frac{3N}{4\pi r^3}. \tag{13}$$

3) Generation of Prototypes: The nonuniform color region can be efficiently represented by the multiscale color prototypes. Given the number of training sample N, maximum and minimum radius of color prototype $r_{\min}$ and $r_{\max}$, the rate of radius

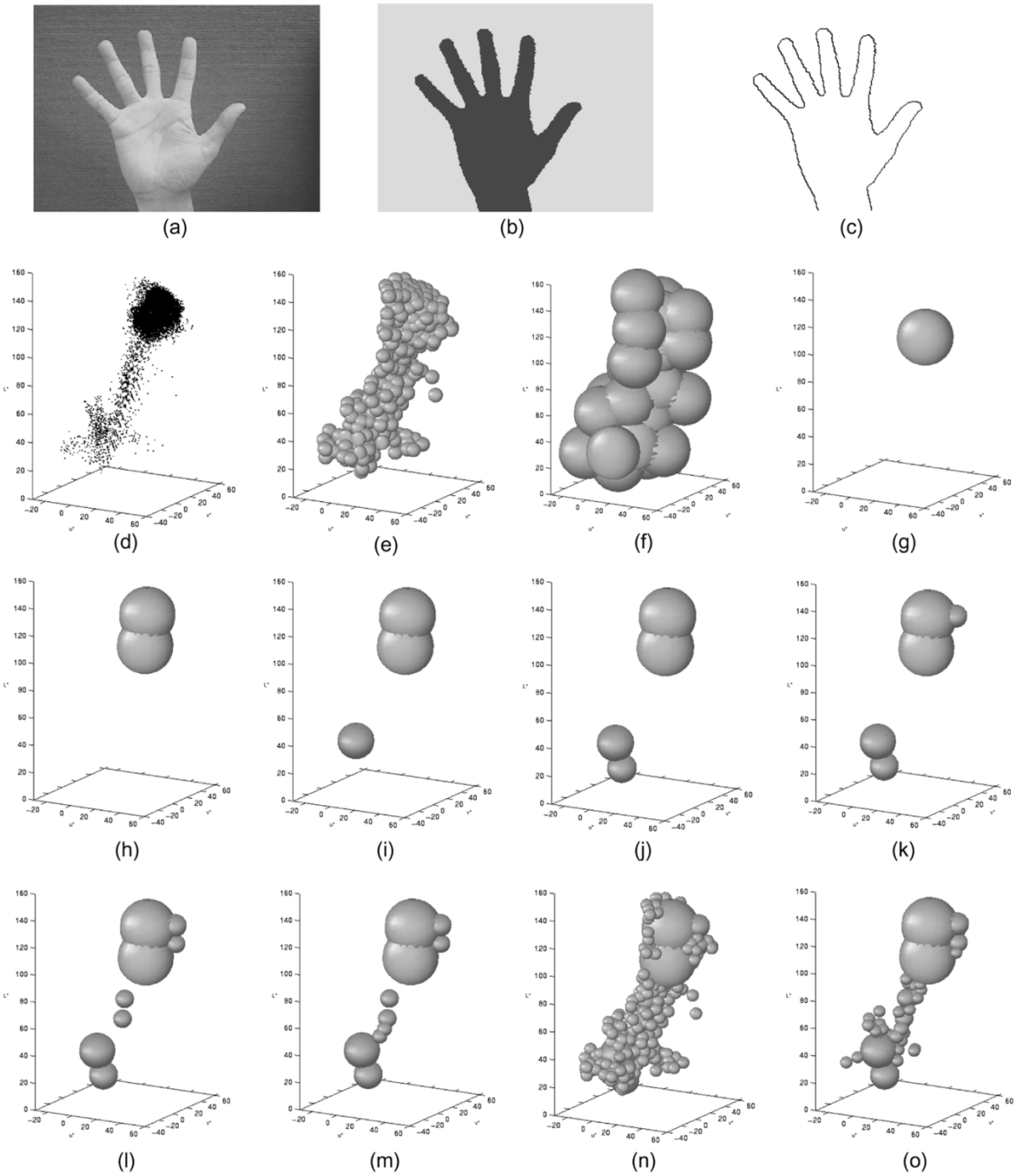

Fig. 5. Hierarchical generation of multiscale color prototypes. (a) Hand gesture image. (b)-(c) Segmentation results using SOM-SA clustering. (d) Hand colors in modified $L^*u^*v^*$ color space. (e)-(f) Color prototype generation with fixed radius $R = 5.0, 20.0$. (g)-(n) Hierarchical generation of multiscale color prototypes, $R_{\min} = 5.0$, $R_{\max} = 20.0$, $\alpha = 0.8$, $D_{\min} = 0.05$. (o) Extraction of representative color prototypes, $N_{\min} = 5$.

descent α, and minimum density of color prototype $D_{\min}$, the multiscale color prototypes can be hierarchically generated as follows:

a) $r = r_{\max}$;
b) *while* $r > r_{\min}$;
c) *for* $i = 1$ *to* N;
d) *if* $\mathbf{x_i}$ is not committed to any of color prototypes, nominate a new color prototype $\mathbf{p}$. The $\mathbf{p}$ is centered at $\mathbf{x_i}$ with the radius r;
e) find the number of color points that are committed to $\mathbf{p}$ in the training sample, compute the D_p;
f) if $D_p > D_{\min}$, accept $\mathbf{p}$, label all committed color points in the training sample;
g) *otherwise*, reject $\mathbf{p}$;
h) *end for*;
i) decrement r by $r_t = \alpha r_{t-1}$;
j) *end while*.

Fig. 5(g)–(n) shows the color prototypes with different radius generated in eight iterations. In the first four iterations, two dominant regions are efficiently enclosed by four large color prototypes. The whole color region is fully enclosed by the multiscale color prototypes in Fig. 5(n).

4) Extraction of Representative Prototypes: Assume the set of color prototypes to be $\mathbf{P} = \{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_m}\}$. In the previous learning procedure, some unrealistic or biased color prototypes may be generated from the noisy colors. It is necessary to remove them from $\mathbf{P}$. Let the committed number of P be $M = \{m_1, m_2, \ldots, m_m\}$. Given the minimum committed number $M_{\min}$, the color prototype $\mathbf{p_i}$ is reserved as a representative prototype if $m_i > M_{\min}$. Otherwise, it is removed from $\mathbf{P}$. By this way, we obtain the set of representative color prototypes $\mathbf{P}^* = \{\mathbf{p_1^*}, \mathbf{p_2^*}, \ldots, \mathbf{p_s^*}\}$ $(s < m)$. Fig. 5(o) shows the representative color prototypes extracted from Fig. 5(n).

C. Pixel Classification

The supervised segmentation can be straightforwardly performed on the test image using $\mathbf{P}^*$. Given the test image $\mathbf{I}$ containing an object of interest (e.g., hand gesture), the pixel $\mathbf{x}$ is classified as the object if it is committed to any $\mathbf{p}^*$ in $\mathbf{P}^*$. Otherwise, $\mathbf{x}$ is regarded as the image background.

V. Experimental Evaluations

The proposed system is a general-purpose tool for the segmentation of color images. The usage of neural networks is able to produce the good segmentation results with a low computational cost. Extensive experiments have been conducted to evaluate the performance of the system. We use the evaluation function in [43] to measure the segmentation results quantitatively, which is formed as follows:

$$Q(I) = \frac{1}{10\,000(N \times M)}\sqrt{R} \times \sum_{i=1}^{R}\left[\frac{e_i^2}{1+\log A_i} + \left(\frac{R(A_i)}{A_i}\right)^2\right] \quad (14)$$

where, $N \times M$ is the image size, R is the number of regions of the segmented image, A_i is the area of the ith region, e_i is the sum of the Euclidean distance between the RGB color vectors of the pixels of region i and the color vector attributed to region i in the segmented image. The $R(A_i)$ represents the number of regions having an area equal to A_i. The experimental evaluations are illustrated with, but not restricted to the following results.

A. Unsupervised Segmentation

The SOM-SA color clustering provides a good tradeoff between the optimal clustering and computational cost. The SA-based color clustering can achieve the best overall segmentation of color images, but it is extremely time consuming. Even with the small size of color image, the computation of annealing process becomes intractably heavy if the SA is directly conducted on the original image. The SOM mapping contributes to the color clustering by two factors: reducing the computational cost and preserving the topology of color clusters. The SA-based color clustering seeks the near-optimal color clusters from a small set of SOM prototypes.

1) Computational Cost: The computational cost can be significantly reduced by the SOM-SA color clustering. Consider that N color points are clustered into K classes by direct SA color clustering, the computational complexity is $O(I_N R_N)$, where I is the number of iterations, and R is the number of color point redistributions. Let M be the number of prototypes in SOM mapping. Then, the computational complexity of SA color clustering becomes $O(I_M R_M)$ in SOM-SA color clustering. Apparently, $O(I_M R_M) \ll O(I_N R_N)$, because $I_M \ll I_N$ and $R_M \ll R_N$.

We have compared the running time of SA and SOM-SA color clustering. The SOM-SA color clustering requires much less time than SA color clustering. To cluster a 116×261 color image into four classes, we run SA and SOM-SA color clustering on the image using a Pentium III 700 MHZ computer. In SA color clustering, we set $R = 30\,000$ and $I = 1000$. It took about 12 h to achieve the whole annealing process. In SOM-SA clustering, we used a 16×16 SOM map, and set $R = 1000$, $I = 500$. The SOM-SA color clustering was accomplished in 19 s. For a large size of color image, the SA color clustering is impractical to use, but the SOM-SA color clustering is still acceptable. We have run the SOM-SA clustering on a 1000×1000 color image, the computation time is about 26 min.

2) Segmentation Results: The two-level approach is used for the color clustering, because SA or SOM alone is unsuitable for color clustering. The SA color clustering is far from the practical usage due to the high computation cost. As stated in [30], SOM mapping is a steepest descent approach, it is prone to being trapped to local optima. In SOM-SA color clustering, SOM mapping only produces the intermediate results for the final SA clustering. The problem of local optima can be greatly alleviated. The SA color clustering is performed on a small set of SOM prototypes with a low computational cost.

We have compared the segmentation results using SOM, SA and SOM-SA color clustering. A comparative result is shown in Fig. 6. The original color image is shown in Fig. 6(a), we segment it into four classes using SOM, SA and SOM-SA color clustering, respectively. In SOM color clustering, we set the size of SOM map to be 2×2. The segmentation result is shown in Fig. 6(b). The SOM color clustering fails to segment the image properly due to the local optima. Fig. 6(c) shows the segmentation result using the SA color clustering. In SOM-SA color clustering, we set the size of SOM map to be 32×32. The segmentation result is shown in Fig. 6(d). In comparison with the SA color clustering, the SOM-SA color clustering can achieve the near-optimal segmentation result.

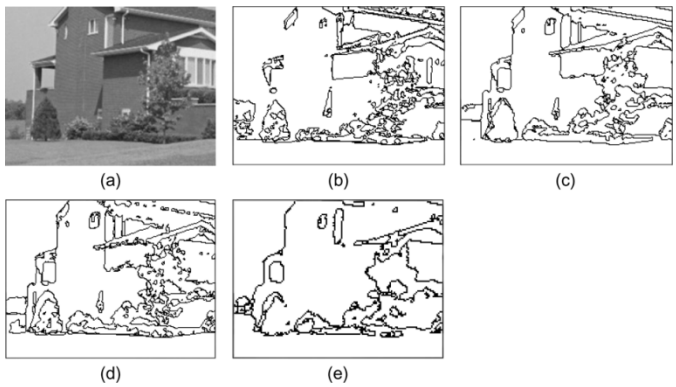

Fig. 6. Quantitative comparison of segmentation results using different color clustering approaches. (a) Original color image. (b) SOM color clustering. $Q = 5580.824$. (c) SA clustering. $Q = 182.526$. (d) SOM-SA color clustering. $Q = 244.826$. (e) CL-SA color clustering. $Q = 376.845$.

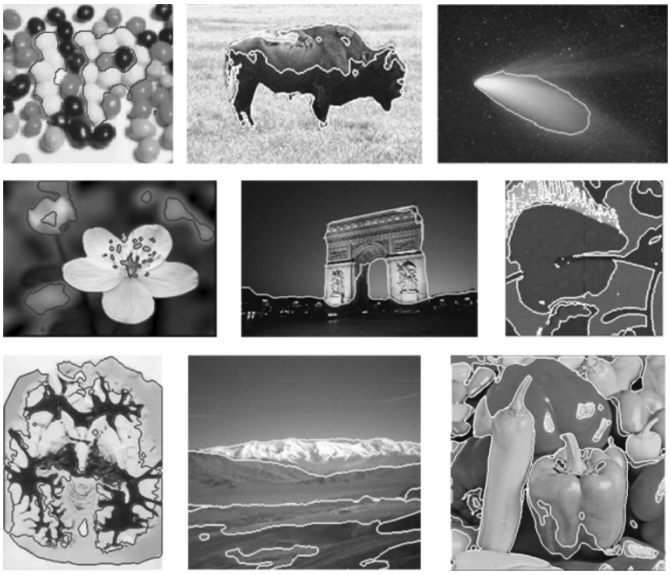

Fig. 7. Segmentation results of some color images using SOM-SA color clustering. The region boundaries are delineated by color curves.

Some two-level clustering approaches have been proposed in the literature, such as CL-Hopfield [19], SOM-cluster merge/discard [22], SOM-agglomerative clustering [25], and SOM-K-means [26]. The CL or SOM is used in the first level. The local clustering approaches are used in the second level. The CL is a common approach for the color reduction [18], [19]. Hence, we have compared the segmentation results using CL-SA and SOM-SA color clustering. The CL-SA color clustering is used to segment the same image in Fig. 6(a). The CL neural networks include 2 layers. In the input layer, three neurons receive the $L^*u^*v^*$ color components. In the output layer, we set 1024 neurons corresponding to 32×32 SOM map in SOM-SA color clustering. The segmentation result is shown in Fig. 6(e). It can found that the SOM-SA color clustering demonstrates the better segmentation result.

3) Some Applications: The SOM-SA unsupervised segmentation has been tested on a variety of color images with the good performance. Some segmentation results are shown in Fig. 7. There are a few parameters needed to be specified by the user for the network training. By a default parameter setting, the SOM-SA color clustering could work well in most cases. However, we have found that the SOM mapping may fail to provide a good color reduction when the size of SOM map is too small. If the computational cost is acceptable, it is advisable to use a large size of SOM map for color reduction.

B. Supervised Segmentation

The HPL learning offers an efficient and accurate way to learn the sample of object colors based on color prototypes. It is especially useful to represent the complex distribution of object colors, whereas the common color learning approaches often fail to represent it properly. We have evaluated the performance of supervised segmentation based on HPL learning. The experimental results are as follows:

1) HPL Learning: We have compared the HPL learning with the common color learning approaches. Fig. 8(a) shows an image of hand gesture that differs from the one in Fig. 5(a). The supervised segmentation is used to segment the hand gesture from the image background. The sample of hand colors is shown in Fig. 7. Fig. 8(b)–(e) shows the results of supervised segmentation using HPL learning, color thresholding, and color histogram. The supervised segmentation using HPL learning demonstrates the best segmentation result.

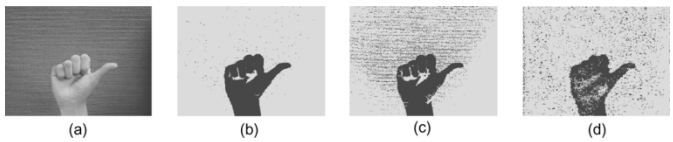

Fig. 8. Segmentation results using different supervised segmentation approaches. (a) Original gesture image. (b) HPL learning. (c) Color threshold. (d) Color histogram.

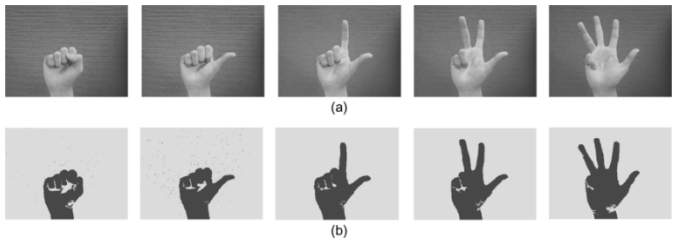

Fig. 9. Supervised segmentation of hand gesture. (a) Original hand gesture images. (b) Segmentation of hand gestures.

2) Some Applications: The proposed supervised segmentation approach is applicable to a broad range of vision tasks. We have used it to segment the object of interest in gesture-based HCI, object tracking, image retrieval, and video surveillance etc. The HCL learning has proven to be a promising way to represent the object colors. Some experimental results are highlighted as follows:

The recognition of hand gestures has become an important part of HCI. In order to recognize the hand gesture from the image sequence, it must be segmented from the image background. Human hand is a nonrigid and moving object in the image. Hence, hand colors are important perceptual features for the segmentation of hand gesture. However, it is difficult to build a generic model of hand colors, because hand colors dramatically change in different lighting conditions [44], [45]. The supervised segmentation provides a good solution for it. We collect the sample of hand colors from the sample image. The hand colors are accurately represented by the color prototypes based on HPL learning. Fig. 9(a) shows the five gesture images in the image sequence. By using the set of representative color prototypes in Fig. 5(o), hand gesture can be properly segmented from the image background. Fig. 9(b) shows the results of hand gesture segmentation.

The *object tracking* is a procedure of state estimation and object segmentation. The state estimation predicts the position of the moving object in the image, and the moving object is segmented in the predicted area. We have applied the HPL-based supervised segmentation for the object tracking. Fig. 10 shows the results of object tracking in two image sequences. The Kalman filter is used for state estimation. The object colors (ping-pong ball and dog) are represented by the color prototypes

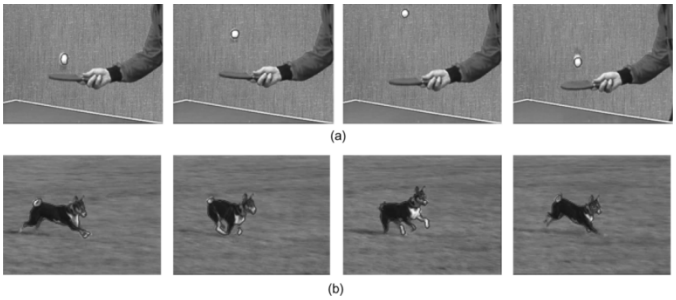

Fig. 10. Object tracking in two image sequences. (a) Tracking of ping-pong ball. The image sequence has 111 frames with the size 320 × 240. The frames 12,13,17,25 are shown. (b) Tracking of dog. The image sequence has 265 frames with the size 320 × 240, The frames 221,225,228, and 232 are shown.

from HPL learning. It can be observed that the moving object is successfully segmented from the image background.

VI. Conclusion

The proposed segmentation system provides a complete solution for both unsupervised and supervised segmentation of color images based on neural networks. The main issues of color image segmentation is systematically addressed in this paper, including perceptual uniformity in color representation, color reduction and clustering in unsupervised segmentation, and color learning in supervised segmentation. In this system, unsupervised segmentation is implemented by SOM-based color reduction, and SA-based color clustering. The supervised segmentation is achieved by HPL learning and pixel classification.

The SOM-SA unsupervised segmentation is able to produce the near optimal segmentation result with a low computational cost. Rather than seeking the color clusters in the original image data, color clustering is carried on the SOM prototypes generated from SOM learning. In color reduction, image colors are projected on a 2-D SOM map, where the structure of clusters is preserved. The SA is used to find the optimal clusters from the SOM prototypes.

The supervised segmentation extends the learning ability of RCE neural network. A new HPL learning procedure is introduced to generate the different sizes of color prototypes. With the further extraction of color prototypes, the distribution of object colors can be efficiently and accurately estimated by the representative color prototypes.

The proposed segmentation system has been applied to some vision tasks with the good results. The SOM-SA unsupervised segmentation has the straightforward application in image analysis and understanding. The natures of topology preserving in SOM and global optimization in SA make the segmentation result robust for the noise image data. The HPL learning and pixel classification are applicable for the supervised segmentation of the object in vision tasks. The HPL learning provides an efficient and accurate way for the estimation of object colors. The object can be properly segmented from the image background based on the color prototypes.

In summary, this segmentation system is a promising tool for the segmentation of color images based on neural networks. The generality of the system makes it applicable in a wide range of computer vision tasks.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments to improve the final manuscript. The access to computing facilities provided by the Institute of High Performance Computing during the implementation of this research work is also kindly acknowledged.

References

[1] L. Lucchese and S. K. Mitra, "Color image segmentation: A state-of-art survey," in *Proc. Indian Nat. Sci. Acad. (INSA-A)*, vol. 67-A, New Delhi, India, Mar. 2001, pp. 207–221.
[2] B. Bhanu, Ed., *Genetic Learning for Adaptive Image Segmentation*. Norwell, MA: Kluwer, 1994.
[3] J. Liu and Y. Yang, "Multiresolution color image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 7, pp. 689–700, Jul. 1994.
[4] S. Ji and H. W. Park, "Image segmentation of color image based on region coherency," in *Proc. IEEE Conf. Image Processing*, Chicago, IL, Oct. 1998, pp. 80–83.
[5] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and byes/mdl for multiband image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, Sep. 1996.
[6] Y. D. B. S. Manjunath and H. Shin, "Color image segmentation," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1999, pp. 1021–1025.
[7] P. E. Trahanias and A. N. Venersanopoulos, "Vector order statistics operators as color edge detectors," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 135–143, Feb. 1996.
[8] M. A. Ruzon and C. Tomasi, "Color edge detection with the compass operator," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 2, Fort Collins, CO, Jun. 1999, pp. 511–514.
[9] D. Androutsos, K. N. Plataniotis, and A. N. Venetsanopoulos, "Distance measures for color image retrieval," in *Proc. IEEE Conf. Image Processing*, vol. 2, Chicago, IL, Oct. 1998, pp. 770–774.
[10] L. Shafarenko, M. Petrou, and J. Kitter, "Histogram-based segmentation in a perceptuall uniform color space," *IEEE Trans. Image Process.*, vol. 7, no. 9, pp. 1354–1358, Sep. 1998.
[11] H. G. Wilson, B. Boots, and A. A. Millward, "Comparison of hierarchical and partitional clustering techniques for multispectral image classification," in *Proc. IEEE Conf. Geoscience Remote Sensing Symp.*, 2002, pp. 1624–1626.
[12] R. H. Turi, "Clustering-based color image segmentation," Ph.D. dissertation, School Comput. Sci. Software Eng., Monash University, Australia, 2001.
[13] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 1–18, May 2002.
[14] J. Shi and J. Malik, "Normalized cuts and image semgmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[15] P. W. Power and R. S. Clist, "Comparison of supervised learning techniques applied to color segmentation of fruit images," in *Proc. SPIE Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Material Handling*, vol. 2904, Boston, MA, Nov. 1996, pp. 370–381.
[16] Y. Qi, A. Hauptmann, and T. Liu, "Supervised classification for video shot segmentation," in *Proc. IEEE Conf. Multimedia and Expo*, vol. 2, Baltimore, MD, Jul. 2003, pp. 689–692.
[17] N. Vandenbroucke, L. Macaire, and K. Postaire, "Color image segmentation by supervised pixel classification in a color texture feature space," in *Proc. IEEE Conf. Pattern Recognition*, vol. 3, Barcelona, Spain, Sep. 2000, pp. 621–624.
[18] T. Uchiyama and M. A. Arbib, "Color image segmentation using competitive learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 12, pp. 1197–1206, Dec. 1994.
[19] P. Scheunders, "A comparison of clustering algorithms applied to color image quantization," *Pattern Recognit. Lett.*, vol. 18, pp. 1379–1384, Nov. 1997.
[20] J.-H. Wang, J.-D. Rau, and W.-J. Liu, "Two-stage clustering via neural networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 606–615, May 2003.
[21] Y. Jiang, K. J. Chen, and Z. H. Zhou, "Som-based image segmentation," in *Proc. 9th Conf. Rough Sets, Fuzzy Sets, Data Mining and Gradnular Computing*, Chongqi, China, May 2003, pp. 640–643.

[22] S. H. Ong, N. C. yeo, K. H. Lee, Y. V. Venkatesh, and D. M. Cao, "Segmentation of color images using a two-stage self-organizing network," *Image Vis. Comput.*, vol. 20, pp. 279–289, 2002.
[23] J. S. Kirk, D. Chang, and J. M. Zurada, "A self-organizing map with dynamic architecture for efficient color quantization," in *Proc. IEEE Int. Joint Conf. Neural Networks*, vol. 3, Washington, DC, Jul. 2001, pp. 2128–2132.
[24] C.-H. Chang, P. Xu, R. Xiao, and T. Srikanthan, "New adaptive color quantization method based on self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 237–249, Jan. 2005.
[25] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 3, pp. 586–600, May 2000.
[26] J. Moreira and L. D. F. Costa, "Neural-based color image segmentation and classification using self-organizing maps," in *Proc. Anais do IX Simpósio Brasileiro de Computacõ Gráfica e Processamento de Imagens (SIBGRAPI'96)*, Caxambu, MG, Brasil, Oct. 1996, pp. 47–54.
[27] M. Porrmann, U. Witkowski, and U. Rückert, "A massively parallel architecture for self-organizing feature maps," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1110–1121, Sep. 2003.
[28] R. Cook, I. McConnell, D. Stewart, and C. Oliver, "Segmentation and simulated annealing," in *Proc. SPIE in Microwave Sensing and Synthetic Aperture Radar*, vol. 2958, G. F. Oliver, Ed., 1996, pp. 30–35.
[29] P. W. Fieguth and S. Wesolkowski, "Highlight and shading invariant color image segmentation using simulated annealing," in *Proc. 3rd Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, INRIASophia-Antipolis, France, Sep. 2001, pp. 314–327.
[30] H. Douzono, S. Hara, and Y. Noguchi, "A clustering method of chromosome fluorescence profiles using modified self-organizing map controlled by simulated annealing," in *Proc. IEEE-INNS-ENNS Joint Conf. Neural Networks*, Como, Italy, Jul. 2000, pp. 4103–4106.
[31] S. Lončarić and Z. Majcenic, "Multiresolution ct head image analysis using simulated annealing," in *Proc. 20th Int. Conf. Information Technology Interfaces*, Pula, Croatia, Jun. 1998, pp. 257–262.
[32] P. Campadelli, D. Medici, and R. Schettini, "Color image segmentation using hopfield networks," *Image Vis. Comput.*, vol. 15, no. 3, pp. 161–166, May 1997.
[33] E. Littmann, "Visual gesture-based robot guidance with a modular neural system," in *Proc. IEEE Conf. Neural Information Processing Systems-Natural and Synthetic*, vol. 8, 1995, pp. 903–909.
[34] A. E. A. N. Papamarkos and C. P. Strouthopoulos, "Adaptive color reduction," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 1, pp. 44–56, Feb. 2002.
[35] T. Riemersma. (1998) Color Metric. [Online]. Available: http://ourworld.compuserve.com/homepages/compuphase/cmetric.htm
[36] C. Poynton. (1999) Frequently asked questions about gamma. GammaFAQ.pdf. [Online]. Available: http://www.inforamp.net/~poynton/GammaFAQ.html
[37] R. W. G. Hunt, Ed., *Measuring Color*, 2nd ed. Chichester, U.K.: Ellis Horwood, 1992.
[38] A. Nemcsics, Ed., *Color Dynamics*. Budapest, Hungary: Akadmiai Kiad, 1993.
[39] K. Matkovic, "Tone mapping techniques and color image difference in global illumination," Ph.D. dissertation, Inst. Comput. Graphics and Algorithms, Vienna Univ. Technol., Wien, Austria, 1997.
[40] T. Kohonen, Ed., *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1995.
[41] D. L. Reilly, L. N. Cooper, and C. Elbaum, "Neural mode for category learning," *Biol. Cybern.*, vol. 45, no. 1, pp. 35–41, 1982.
[42] P. E. H. R. O. Duda and D. G. Stork, Eds., *Pattern Classification*, 2 ed. New York: Wiley, 2001.
[43] M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative evaluation of color image segmentation results," *Pattern Recognit. Lett.*, vol. 19, pp. 741–747, 1998.
[44] Y. Wu and T. S. Huang, "Hand modeling, analysis, and recognition," *IEEE Signal Process. Mag.*, vol. 18, no. 3, pp. 51–60, May 2001.
[45] R. Kjeldsen and J. Kender, "Finding skin in color images," in *Proc. IEEE Conf. Automatic Face and Gesture Recognition*, Killington, VT, Oct. 1996, pp. 312–317.

Guo Dong (M'03) received the B.Sc. degree in electronic precision machines from the University of Electronic Science and Technology of China, Chengdu, China, in 1987 and the M.Sc. and Ph.D. degrees in robotics and automation from Tianjin University, Tianjin, China, in 1993 and 1997, respectively.

From 1998 to 1999, he was a Research Fellow in the Robotics Research Center, Nanyang Technological University, Singapore. In 2000, he was a Postdoctoral Researcher with INRIA, Rhône-Alpes, France. He is an Associate Editor of the *International Journal of Humanoid Robotics*. He is currently a Senior Member of the Technical Staff at DSO National Laboratories, Singapore. His current research interests include image segmentation and grouping, neural networks, machine learning, and human–robot interaction.

Ming Xie (M'91) received the M.Sc. degree from the University of Valenciennes, Valenciennes, France, in 1986 and the Ph.D. degree from the University of Rennes, Rennes, France, in 1989, for his research works done at INRIA.

He is currently an Associate Professor (on tenure) with the Nanyang Technological University (NTU), Singapore. He has completed many funded academic and industrial projects, and is still undertaking original research investigations in the important fields of humanoid robotics, such as manipulation, biped walking, cognitive vision, and natural language understanding. He is the author of a popular textbook entitled *Fundamentals of Robotics* (Singapore: World Scientific). Since 2004, he is one of the Editors-in-Chief of the *International Journal of Humanoid Robotics*. He has contributed to the Singapore-MIT alliance for postgraduate programs in science and engineering. And, before taking up the faculty position, he has worked with, and for, Renault Automation Ltd., Renault Automobile Co., and a private IT company in Paris, France.