

Effects of Classification Methods on Color-Based Feature Detection With Food Processing Applications

Kok-Meng Lee, *Fellow, IEEE*, Qiang Li, and Wayne Daley

Abstract—Color information is useful in vision-based feature detection, particularly for food processing applications where color variability often renders grayscale-based machine-vision algorithms that are difficult or impossible to work with. This paper presents a color machine vision algorithm that consists of two components. The first creates an artificial color contrast as a prefilter that aims at highlighting the target while suppressing its surroundings. The second, referred to here as the statistically based fast bounded box (SFBB), utilizes the principal component analysis technique to characterize target features in color space from a set of training data so that the color classification can be performed accurately and efficiently. We evaluate the algorithm in the context of food processing applications and examine the effects of the color characterization on computational efficiency by comparing the proposed solution against two commonly used color classification algorithms; a neural-network classifier and the support vector machine. Comparison among the three methods demonstrates that statistically based fast bounded box is relatively easy to train, efficient, and effective since with sufficient training data, it does not require any additional optimization steps; these advantages make SFBB an ideal candidate for high-speed automation involving live and/or natural objects.

Note to Practitioners—Variability in natural objects is usually several orders of magnitude higher than that for manufactured goods and has remained a challenge. As a result, most solutions to inspection problems of natural products today still have humans in the loop. One of the factors influencing the success rate of color machine vision in detecting a target is its ability to characterize colors. When unrelated features are very close to the target in the color space, which may not pose a significant problem to an experienced operator, they appear as noise and often result in false detection. This paper illustrates the applicability of the algorithm with a number of representative automation problems in the context of food processing applications. As demonstrated experimentally, the artificial color contrast and statistically based fast bounded box methods can significantly improve the success rate of the detection by reducing the standard deviation of both the target and noise pixels, enlarging the separation between feature clusters in color space, and more tightly characterize the feature color from its background. The algorithm presented here has several advantages, including simplicity in training and fast classification, since only three simple checks of rectangular bounds are performed.

Index Terms—Color classification, feature detection, inspection, machine vision (MV).

Manuscript received July 19, 2005; revised November 7, 2005. This paper was recommended for publication by Associate Editor Y. Li and Editor M. Wang upon evaluation of the reviewers' comments.

K.-M. Lee and Q. Li are with the The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405 USA (e-mail: kokmeng.lee@me.gatech.edu).

W. Daley is with the Food Processing Division, Georgia Tech Research Institute, Atlanta, GA 30332 USA.

Color versions of Figs. 2, 3, 7, 8, 10, 12, 14, 15, and 17 are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TASE.2006.874972

I. INTRODUCTION

NATURAL object identification has received more attention in automation. Early works focused on identifying a human face from grayscale images [1]–[4] using edge and shape information. More recently, color vision as an image processing tool in detecting features has widely been adapted to human face identification [5]–[8]. Color-based algorithms have been found to be much faster than those based on shape. However, color vision has its problems when used in natural object identification since color images of natural products are susceptible to noise. A good lighting system can only partially solve the noise problem. In addition, color variation and uniformity are a unique nature of live objects. The design of a time-efficient, reliable color classification algorithm for food processing applications has remained a challenge.

Machine vision (MV) algorithms for detecting features of a moving natural object can be classified into three main categories: 1) features are extracted from a gray-level image on the basis of both edge and shape; 2) features are detected by virtue of their characteristic color; 3) the method uses a combination of 1) and 2); often characteristic color detection is applied as a preprocessor followed by a shape-matching algorithm to identify the target feature. Among them, characteristic color detection has been an attractive solution particularly in applications where the color difference between the target features and its background is significant. For high-speed automation applications [9], such as food processing and handling of live objects for meat production, the shape variation and voluntary motion of live, natural objects coupled with the stringent production demands to reduce computation time make algorithms based on shape information less than attractive for real-time applications. Color-based algorithms are especially suitable for detecting features of live natural objects and have significant potential in agriculture and food processing. The basic idea of a color-based feature detection algorithm is to utilize a set of training data to approximate the boundary of the color subspace that characterizes the feature for subsequent classification. Reilly–Cooper–Elbaum's neural-network classifier (RCE–NNC) [10] has been one of the most commonly used methods for identifying shape patterns [3], [11]–[15]. It has also been used in color vision [16] and orientation detection in handling live birds [17]. The performance of RCE–NNC as compared to other neural networks has been studied in [18]. The success rate of an RCE–NNC depends heavily on its designed parameters and the topology of the trained network. Although some research effort (for example, [13] and [19]) has been directed toward optimizing the topology of an RCE–NNC, its optimization for a real-time application has remained a

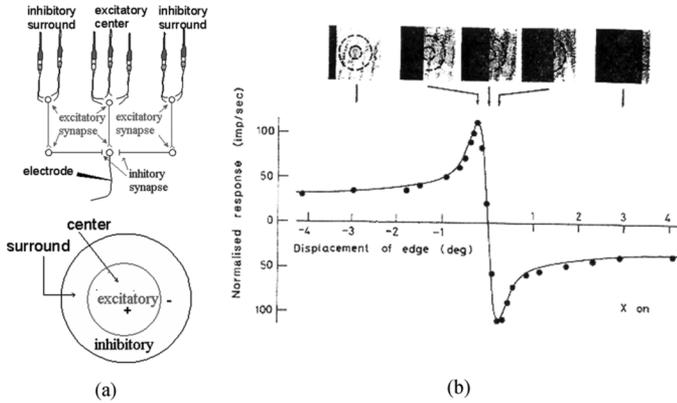


Fig. 1. Receptive field and ganglion cell. (a) ON-OFF-center. (b) Cat's RGC response to an edge stimulus [26].

challenge. More recently, kernel methods have become more popular [20]. As one of the commonly used kernel methods, the support vector machine (SVM) [21]–[23] has been developed from a rigorous statistical learning theory. Both RCE–NNC and SVM are relatively easy to use compared to many other NNCs. In practice, both methods, however, could become less efficient and inaccurate when their parameters are poorly tuned; the procedure for optimizing the design parameters could be tedious and time-consuming.

To improve the success rate of detecting color-based features involving natural or live products, color vision algorithms must be able to discriminate target features from noise at high speed. For these reasons, our color vision research has focused on addressing two specific issues: The first is to create artificial color contrast (ACC) as a prefilter to highlight the target but suppress its surroundings. The second is to improve the characterization of the color feature in an attempt to exclude noise so that the color classification can be performed more accurately and efficiently.

Human abilities to perceive fine gradation of color have motivated us to utilize some functions in a human visual system (HVS) to address the first issue encountered in automating visual inspection currently performed by human inspectors in a production setting. In early 1953, Kuffler [24] combined the use of microelectrodes for recording directly from retinal ganglion cells (RGC) bodies in the intact eye of the cat with the simulation of the retinal with localized spots of light. He found two kinds of ganglion cells, ON- and OFF-center as shown in Fig. 1(a), for those excited and inhibited by light in the center of their receptive fields, respectively. Many of the basic anatomical and physiological principles evident in the cat retina also hold for the primate retina [25]. Built upon Kuffler's method, Enroth–Cugell and Rodson [26] used the difference of two Gaussian (*DoG*) functions to quantitatively describe the RGC behaviors. Their modeled data reproduced in Fig. 1(b), which shows the striking ability of an HVS to emphasize edges while simultaneously smoothing its surroundings and maintaining the level of contrast, has been the basis for visual psychophysics and computer vision work addressed to edge detection and image segmentation. In this paper, we explore this

ability for color machine vision formulated using trichromacy and opponent-color theories to separate closely similar color features in handling natural objects and with the inspection of food products; the opponent-color theory was qualitatively hypothesized by Hering [27] to explain various phenomena that could not be adequately accounted for by trichromacy.

The remainder of this paper offers the following:

- 1) We present an alternative method, the statistically based fast bounded box (SFBB), to characterize the color features. SFBB uses the principal component analysis (PCA) technique [28] to obtain the principal axes of the training data distribution in the color space. PCA is well known for its use in the eigenface algorithm [1], [2] that helps find the most dominant feature from a grayscale image of a human face. Unlike [1] and [2] where PCA was used to reduce the dimension of the identification problem with the whole grayscale image as a classification input, the SFBB method finds a linear transformation to minimize the covariance of the training set where individual color pixels are used as classification inputs.
- 2) We provide a general formulation that uses *DoG* with opponent colors to increase the separation between target features and noise in color space so that unrelated features (or noise) can be easily excluded from the bounded box characterizing the target. This approach utilizes some functions in the human visual system (HVS) to create ACC for improving the reliability in finding color features in a natural product. However, we also recognize that HVS does not exhibit perfect color constancy [29], [30] and also performs poorly in lighting with abnormal spectral content (e.g., sodium arc) particularly when color features are very similar. Thus, we develop here a robust method utilizing the quantitative ability of a color machine vision to discriminate a very small color difference between similar features, and show how Hering's theory of opponent colors can be quantified in designing color-based vision algorithms. Unlike color constancy that refers to the lack of change in the perceived color of a colored patch as the global illumination changes, we focus on contrast due to the change in perceived color of a colored patch as its local surrounding is changed, given the illumination.
- 3) We illustrate the use of SFBB with ACC for characterizing color-based features in the context of food processing applications involving live products or biomaterials, where reliability and high processing speed are of particular concern. In general, variability in these products is usually several orders of magnitude higher than that for manufactured goods.
- 4) We examine the effects of the color boundary on the computational efficiency of feature detection by comparing three color classification algorithms; namely, SFBB, RCE–NNC [10] [17], and SVM [21]–[23] which are chosen from the viewpoint of the geometrical boundary. SFBB uses a bounded box with orientation and dimensions defined by the statistics of the training set while RCE–NNC uses hyperspheres and SVM uses hyperplanes. The visual comparison offers some intuitive insights on the effect of their differences on the success rate of detection.

II. STATISTICALLY BASED FAST BOUNDED BOX (SFBB)

A color signal is represented here in red, green, and blue (RGB) space, and the target is considered as a subspace in the entire RGB color space.

A. Problem Formulation

The identification problem can be defined as follows: Given a set of scatter points (referred to here as a training set) in the target color subspace Ω , the problem is to find its boundary Γ such that for any color vector \mathbf{C} in the color space, if \mathbf{C} is inside Γ , then \mathbf{C} represents one point of the target feature; otherwise, it does not belong to the feature.

If all of the members in Ω are known, the boundary of Ω is also known. Unfortunately, the boundary of the target color subspace can only be constructed from a limited set of training samples; thus, it is essentially an approximation at best. The closeness and shape of the approximated boundary, which depends on the decision rules of the specific algorithm employed, have a significant influence on the cycle time and success rate of the automated process.

B. Simple Bound Box in RGB Space and Its Problems

The interest here is to describe the boundary Γ that simplifies the subsequent identification process and makes the detection algorithm accurate and fast. A simple bounded box can be used to approximate the boundary of the target color subspace. This method assumes that the three color component vectors are independent random variables. The basic idea here is to construct a smallest possible rectangular box to enclose representative color points of the target from a given training set in RGB space. Such a bounded box can be easily constructed from the maximum and minimum values of the color component. The classification can then be reduced to simply check whether the RGB pixel values are within the bounds, which is a relatively simple and fast process.

The above procedure, however, does not result in the tightest box since it does not take into account the color characteristics of the feature. In fact, this larger-than-necessary box would result in introducing unwanted color points (as noise) in the processed image.

A relatively simple method to examine whether the three color component vectors are independent random variables is to fit the normal distribution to the training data. If the training set matches the ideal normal distribution very well, there is a very high probability that these components are independent of each other. Otherwise, some of them may be related and a change of the variables is necessary to minimize correlation among the variables.

C. Finding SFBB Using Principal Component Analysis

In order to minimize the correlation among the component vectors, the SFBB method computes the three principal axes of the training set for constructing the bounded box. The method involves the following steps.

- First, the principal axes characterizing the training set are calculated from its covariance matrix.

Given training set \mathbf{X} with component vectors \mathbf{X}_r , \mathbf{X}_g , and \mathbf{X}_b in the RGB space, a covariance matrix $[\mathbf{U}]$ can be computed from (1)

$$[\mathbf{U}] = \begin{bmatrix} \sigma^2\{\mathbf{X}_r\} & \sigma\{\mathbf{X}_r, \mathbf{X}_g\} & \sigma\{\mathbf{X}_r, \mathbf{X}_b\} \\ \sigma\{\mathbf{X}_g, \mathbf{X}_r\} & \sigma^2\{\mathbf{X}_g\} & \sigma\{\mathbf{X}_g, \mathbf{X}_b\} \\ \sigma\{\mathbf{X}_b, \mathbf{X}_r\} & \sigma\{\mathbf{X}_b, \mathbf{X}_g\} & \sigma^2\{\mathbf{X}_b\} \end{bmatrix} \quad (1)$$

where

$$\sigma^2\{\mathbf{X}_i\} = \frac{1}{N-1} \sum_{n=1}^N (X_{in} - \bar{X}_i)^2 \quad i \in \{r, g, b\} \quad (1a)$$

$$\sigma\{\mathbf{X}_i, \mathbf{X}_j\} = \frac{1}{N-1} \sum_{n=1}^N (X_{in} - \bar{X}_i)(X_{jn} - \bar{X}_j) \quad i, j \in \{r, g, b\}; i \neq j \quad (1b)$$

$$\bar{X}_i = \frac{1}{N} \sum_{n=1}^N X_{in} \quad (1c)$$

and N is the number of color points. The covariance matrix $[\mathbf{U}]$ is symmetric. To maximize the variances of interest, we apply the singular-value-decomposition method to obtain the eigen values ($\lambda_1, \lambda_2, \lambda_3$) and the normalized eigen vectors ($\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$) of $[\mathbf{U}]$. The three principal axes of the new coordinate system are given by the unit vectors along $\mathbf{v}_1, \mathbf{v}_2$, and \mathbf{v}_3 .

- Second, the training set in RGB space is transformed to the new coordinate system formed by the three principal axes. The training data can be transformed from its original RGB space to the new coordinate system

$$\hat{\mathbf{X}} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3]^T (\mathbf{X} - \bar{\mathbf{X}}) \quad (2)$$

where $\mathbf{X} = [X_r \ X_g \ X_b]^T$ and $\bar{\mathbf{X}}$ are the original vector and its corresponding average in RGB space, respectively, and $\hat{\mathbf{X}} = [X_1 \ X_2 \ X_3]^T$ is the transformed color vector.

- Finally, the bounded box is constructed in the new coordinate system.

In order to find an appropriate size for the bounded box to best characterize the color of the feature, we use linear regression theory to determine the confidence level at which the three transformed color components are independent random variables; this level is used to specify the boundary of the box in terms of standard deviations (SD).

D. Illustrative Example: Live-Bird Handling Application

We illustrate the computation procedure with a live-bird handling application for meat processing, where the birds must be shackled in a specific direction. The bird's orientation (forward/backward) is determined by identifying its red comb relative to its white-feathered body as shown in Fig. 2. Due to varying sizes and shapes, and some natural reflexes (or voluntarily motion) of the birds, MV algorithms based on detecting edges and/or shapes have difficulties meeting stringent production requirements that demand reliability and speed. In this application, the vision algorithm is used to detect the red comb of a bird. Apart from a spectrum of red that characterizes the combs in a typical batch, noise (such as dirt on the feathers, bare spots of flesh,

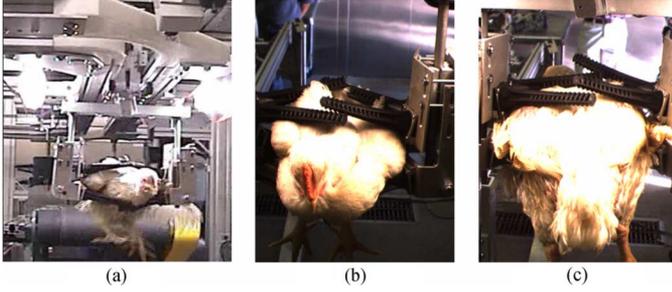


Fig. 2. Experimental setup and typical images. (a) Setup. (b) Forward. (c) Backward.



Fig. 3. Example illuminated combs of typical birds.

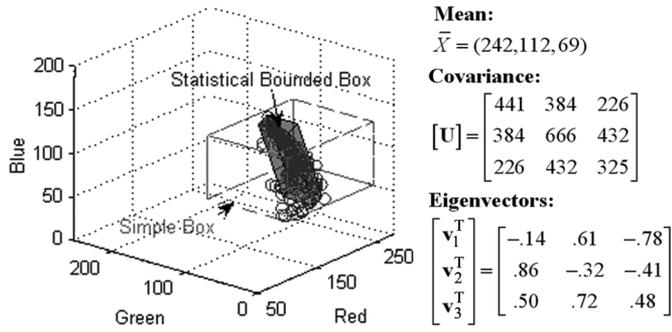


Fig. 4. Training set and the computed SFBB, data from Fig. 3.

shadow and reflection of environmental illumination) present a challenge to reliable color detection as shown in Fig. 3.

Fig. 4 shows a set of experimentally obtained training data of N color points from the comb. The data are plotted in RGB space, along with corresponding covariance matrix and normalized eigenvectors, where the rectangular box represents the computed SFBB (with dimensions at ± 2 SD or 95% confidence level). Fig. 5 compares the normal distribution of the red components of the training set against those of the transformed vector components. In Fig. 5, the distribution is graphed in a natural logarithmic scale; the dashed line is the ideal normal distribution and the discrete points are experimentally obtained data.

As shown in Fig. 5, the original red component data do not match the normal distribution and, thus, its vector is not an independent random variable. The data distribution in transformed coordinates matches the normal distribution very well; the transformed coordinates can be treated as independent random variables. This justification permits the boundary of the bounded box to be specified in terms of the number of standard deviations (SDs) of each component. Clearly, the tightest box (that more closely characterizes the color of the comb) is preferred since

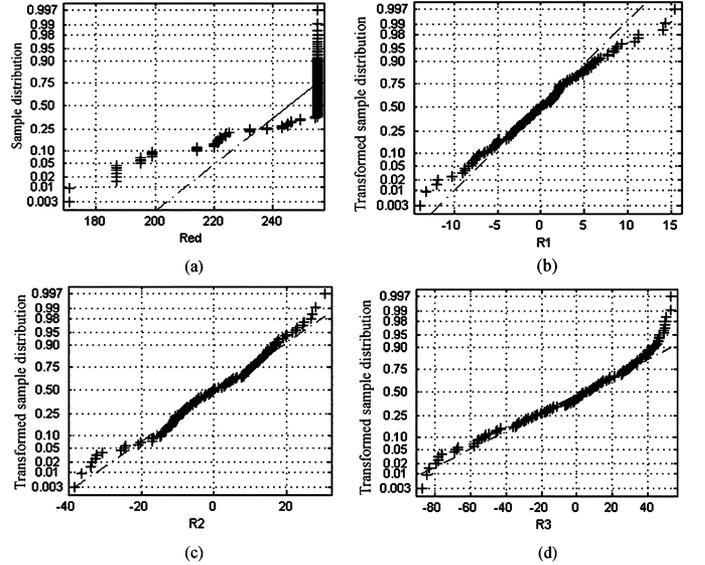


Fig. 5. Normal distribution (before and after transform). (a) Red color. (b) R1 (transformed coordinate). (c) R2 (transformed coordinate). (d) R3 (transformed coordinate).

a larger-than-necessary box would include unrelated features as noises, which must be excluded before color classification. Once SFBB is obtained, the classification becomes a straightforward process of checking whether the test color pixels fall within the bounded box.

III. ARTIFICIAL COLOR CONTRAST (ACC)

When unrelated features are very close to the target in color space, which may not pose a problem to an experienced operator, they appear as noise and often result in false detection. In this section, we present a method (ACC prefilter) to separate targets and unrelated features in color space. The method is based on some observations in [24], [26], and [27] that the response function of an HVS is essentially a result of two independent mechanisms (referred to here as center and surround similar to the ON- and OFF-centers of the RGC in an HVS, respectively). This response function is quantitatively modeled as DoG of the center and surround. To account for the effect of colors, we design the center and surround responses based on a combination of the trichromatic and opponent theories of vision.

A. Model for ACC

Consider a 2-D symmetric (zero-mean uncorrelated) Gaussian kernel

$$G_\sigma(x, y) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (3)$$

to get

$$g(x, y) = G_\sigma * f(x, y) \quad (4)$$

where, for simplicity, G_σ denotes $G_\sigma(x, y)$. Similar to Laplace of a Gaussian, the image is first smoothed with the Gaussian

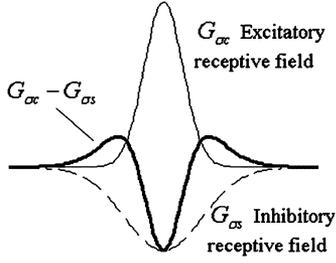


Fig. 6. *DoG* model of receptive field.

kernel of width σ . The difference of two Gaussian-smoothed images can be written as

$$h_i(x, y) = G_{\sigma_c} * f_i(x, y) - G_{\sigma_s} * f_j(x, y). \quad (5)$$

- *DoG* as an edge finder of a single grayscale image. The difference of two Gaussians is commonly used in MV to detect edges in a grayscale image, for which

$$f_i(x, y) = f_j(x, y) = f(x, y).$$

Thus

$$h(x, y) = (G_{\sigma_c} - G_{\sigma_s}) * f(x, y) = DoG * f(x, y) \quad (6)$$

where the subscripts “c” and “s” denote the center and surround of the excitatory and inhibitory receptive fields (Fig. 6), respectively; $\sigma_c < \sigma_s$; and *DoG* as a convolution kernel is defined as

$$DoG \triangleq G_{\sigma_c} - G_{\sigma_s} \\ = \frac{1}{\sqrt{2\pi}} \left[\frac{1}{\sigma_c} e^{-(x^2+y^2)/2\sigma_c^2} - \frac{1}{\sigma_s} e^{-(x^2+y^2)/2\sigma_s^2} \right]. \quad (7)$$

To facilitate the following discussion, we broadly divide the surrounds into two types:

Type I)

$$h_i(x, y) = DoG * f_i(x, y). \quad (8a)$$

Type II)

$$h_i(x, y) = G_{\sigma_c} * f_i(x, y) - G_{\sigma_s} * f_j(x, y). \quad (8b)$$

Type I is essentially an edge detection filter applied on a color component image. Type II allows for $f_i(x, y) \neq f_j(x, y)$. In Types I and II, $f_i(x, y)$ with $(i = 1, 2, 3)$ corresponds to *RGB* component images, respectively; and $f_j(x, y)$ with $(j = 4, 5, 6)$ are some linear combinations of *RGB* component images to be designed. The ACC method is to find

$$f_j(x, y) = \sum_{i=1}^3 \alpha_i f_i(x, y) \quad (9)$$

where α_i are weighting factors.

- *Opponent-Color-Based Surrounds*

We mimic the HVS by denoting the component $f_i(x, y)$ as the center and the linear combination as a surround. For simplicity, we denote the component images as

$$R = R(x, y), \quad G = G(x, y) \text{ and } B = B(x, y).$$

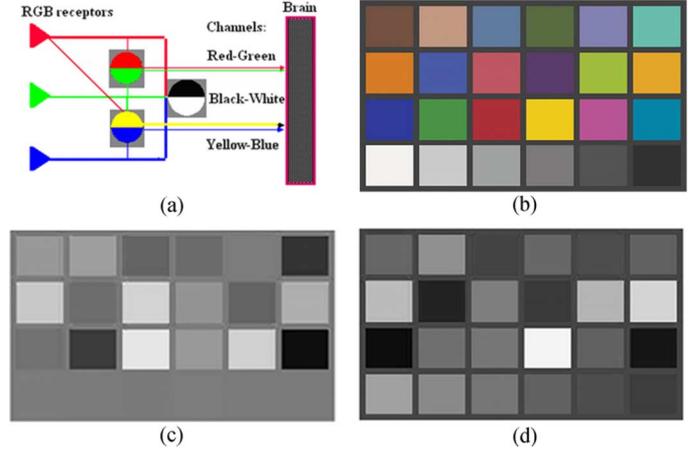


Fig. 7. Effect of R-G and Y-B channels on monitor color test patterns. (a) Hering's theory of opponent color. (b) Color checker. (c) R-G of (a). (d) $R + G - B$ of (a).

One possible set of surrounds (inspired by Herring's opponent color theory) is the $R - G$ and $Y - B$ channels

$$f_{\pm(R-G)}(x, y) = \pm (R - G) \quad (10a)$$

$$f_{\pm(Y-B)}(x, y) = \pm (R + G - B) \quad (10b)$$

where \pm denotes that the channel is bipolar. The term “bipolar” means that each channel can signal only one of the two attributes. Opponent color exists because the sensations of red and green are encoded in a single visual pathway and, similarly, for blue and yellow—an accepted fact in modern biological vision. The effect of the positive channel of (10a) and (10b) on a color calibration pattern is shown in Fig. 7. Much like Hering's opponent cells, $f_{R-G}(x, y)$ excites red and inhibits green. Similarly, $f_{Y-B}(x, y)$ excites yellow and inhibits blue.

Substituting $f_j(x, y)$ into (8b) with (10a) with $+(R - G)$

$$h_1(x, y) = DoG * R + G_{\sigma_s} * G \quad (11-1)$$

$$h_2(x, y) = DoG * G + G_{\sigma_s} * [2G - R] \quad (11-2)$$

$$h_3(x, y) = DoG * B + G_{\sigma_s} * [B - (R - G)]. \quad (11-3)$$

And similarly with (10b) with $+(Y - B)$

$$h_4(x, y) = DoG * R + G_{\sigma_s} * [B - G] \quad (12-a)$$

$$h_5(x, y) = DoG * G + G_{\sigma_s} * [B - R] \quad (12-b)$$

$$h_6(x, y) = DoG * B + G_{\sigma_s} * [2B - (R + G)]. \quad (12-c)$$

In (11) and (12), each transformed component consists of two parts. The first part is essentially a monochrome *DoG* filter allowing the detection of edges in the image. The second part emphasizes the influences of certain color components in order to create the necessary contrast. Similar equations can be obtained using the negative channel of (10a) and (10b).

- *Artificial Color Contrast for Feature Discrimination*

The ACC method is best illustrated with examples. Consider three color vectors (target, noise, and background) in an image, where the target and noise are very close in color space (particularly the red component). The interest is to separate the distance between the target and noise, and that

TABLE I
EFFECT OF DIFFERENT CENTER AND SURROUND COMBINATIONS

Comb.	Color vectors [target] [noise] [background]	Distances
Original	[170 110 108], [169 90 81], [223 210 100]	33.6, 113
1-2-3	[110 50 48], [90 11 2], [210 197 87]	63.5, 182
1-2-c	[110 50 -64], [90 11 -97], [210 197 -233]	54.9, 245
1-b-3	[110 -62 48], [90 -88 2], [210 -123 87]	56.4, 123
1-b-c	[110 -62 -64], [90 -88 -97], [210 -123 -233]	46.5, 205
a-2-3	[-2 50 48], [-9 11 2], [-110 197 87]	60.7, 186
a-2-c	[-2 50 -64], [-9 11 -97], [-110 197 -233]	51.6, 248
a-b-3	[-2 -62 48], [-9 -88 2], [-110 -123 87]	53.3, 130
a-b-c	[-2 -62 -64], [-9 -88 -97], [-110 -123 -233]	42.9, 209

between the target and background. Table I shows eight (2^3) different center/surround combinations that can be created from the positive channel of (10a) and (10b) on their effect on the two distances; the legend “1-2-c” denotes that the transformed components are computed by (11-1), (11-2), and (12-c). Clearly, another eight transformed images can be created from the negative channel of (10a) and (10b).

As illustrated in Table I, both target and noise features have the same gray level in the dominant red of 170. Thus, we create ACC (between target and noise) by taking advantages of a larger difference in the G gray level. This is accomplished by applying the $+(R - G)$ color opponent to the R component image replacing the gray levels in R with that in G as shown in the second part of (11-1) via the surround mechanism. Similar amplification can be achieved by applying the $+(R - G)$ color opponent to the G component image. As shown in Table I, all of these combinations result in a larger distance between the target and noise features in the transformed space; this would reduce the chance of including noise in the bounded box.

The above procedure is equally applicable to other closely similar colors. As an illustration, consider the same example in Table I where the target and background have a similar B gray level of about 100. ACC (between the target and background) can be created by applying the $-(Y - B)$ color opponent as surround to the B component image. The response has the form $DoG * B + G_{\sigma_s} * Y$; the gray levels of B are replaced by those of Y while the edge information in B is preserved. Using the above with (11-1) and (11-2) for the R and G components as discussed earlier, the vectors characterizing the features in the transformed color space are [110, 50, 160] [90, 11, 101] and [210, 197, 507] and the corresponding two distances are 73.5 and 390, which are significantly larger than those of the original image.

The color image is often scaled to 3 bytes for visual presentation in RGB. Such scaling, however, is generally not necessary for machine applications in real time.

B. Illustrative Examples

Due to the page limit, we primarily illustrate food-processing applications here, where red is often a dominant color to be identified. Without loss of generality, only the positive channel of (10a) and (10b) will be used from here on. We expect that the extension to a more complete set of center/surround combinations or for other colors, though lengthy, is straightforward. Based on the results in Table I, we choose the transformed space “1-2-c”

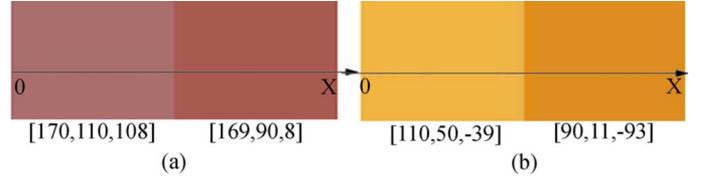


Fig. 8. Example illustrating the basic concept of ACC. (a) Original RGB of image. (b) Transformed ACC image.

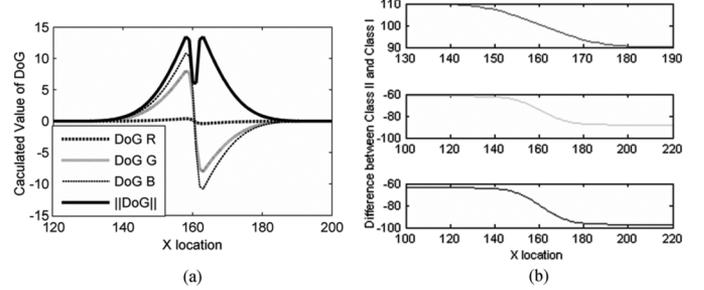


Fig. 9. Effect of ACC on example colors in Fig. 8. (a) Class I: DoG of component color. (b) Type II - Type I.

for clarity of image presentation to general readers. Mathematically, this transformed ACC space is given by (13)

$$\begin{aligned} h_R(x, y) &= h_1(x, y) \\ &= DoG * R + G_{\sigma_j} * G \end{aligned} \quad (13a)$$

$$\begin{aligned} h_G(x, y) &= h_2(x, y) \\ &= DoG * G + G_{\sigma_s} * [2G - R] \end{aligned} \quad (13b)$$

$$\begin{aligned} h_B(x, y) &= h_6(x, y) \\ &= DoG * B + G_{\sigma_s} * [2B - (R + G)]. \end{aligned} \quad (13c)$$

As will be shown, (13) greatly enhances the ability to separate closely similar red features in the transformed background of yellow. The widths $\sigma_c = 1$ and $\sigma_s = 10$ pixels, are used in the following examples.

Example 1: ACC's Effect on Contrast and Edge: Consider the test image shown in Fig. 8(a), which is made up of two similar red features with their RGB values listed below the colors. Fig. 8(b) shows the transformed ACC image, where an offset of 100 is added to the transformed image to allow for visual illustration. Figs. 9(a) and (b) plot the first and second terms of (13) across the edge of the transformed image shown in (8b). Two observations are worth mentioning.

- 1) The ACC method has a significant effect on enhancing contrast (denoted by the ratio $\Delta I/I$ where ΔI is the Euclidean distance between the two features), which increases from 0.15 to 0.57.
- 2) Type II response, as given by the component sum in (8a) and (8b), resembles the cat's RGC response to an edge stimulus shown in Fig. 1 [26].

Example 2: Effect of ACC to Color Classification: Fig. 10(a) shows an image of a white-feathered broiler (meat chicken) on a moving conveyor, where the red comb is to be identified and the featherless spot with shadow is potential noise. A representative set of training data for classification, where two clusters (denoted by red and blue) are color pixels of the target and noise in RGB color space, is given in Fig. 11(a). These clusters (both dominant in red) are very close to each other in the color space; it makes color-based identification a difficult task. Thus, we apply



Fig. 10. Example illustrating the effect of ACC on features of similar color. (a) Original image. (b) Transformed ACC image.

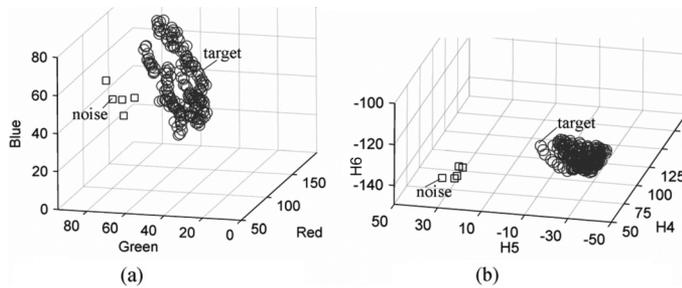


Fig. 11. Comparison showing the ACC's effect on features in color space. (a) Color pixels in RGB space. (b) Color in transformed space.

	Original: [target]/[noise]	Transformed: [target]/[noise]
Mean	[152, 58, 45]/[135, 85, 39]	[61, -34, -123]/[81, 29, -143]
SD	[3.9, 7.2, 22.2]/[1.1, 4.3, 10]	[3.5, 4.2, 12.8]/[0.5, 1.9, 7.5]
Distance	36	69

the ACC method to artificially increase the separation between the two clusters. The transformed image computed using (13) is given in Fig. 10(b).

The effect of ACC on color representation can be illustrated by comparing the pixel clusters of the original RGB image and those of the transformed ACC image in Figs. 11(a) and (b), respectively, along with their means, standard deviation (SD), and distance in Table II, where the SD is calculated along the principal axes. The distance between the clusters in ACC space is twice that of the RGB space. In addition, the application of the Gaussian smoothing filters results in more closely packed clusters. Both of these effects will ease the design of the classifier.

Example 3: Food Processing Applications: We examine the robustness to ACC to sensor resolution in bone detection—an application of great importance to many producers because of food safety concerns. Figs. 12(a) and (b) show two images of a poultry (breast-butterfly) meat taken using two different color cameras:

- 1) high-resolution 3-CCD camera (Sony DXC900);
- 2) low-cost single-chip camera (point gray firefly).

In Fig. 12(a), a “fan-bone” to be identified can be seen at the lower right corner on the surface. In addition, there are blood stains near the fan-bone and on the opposite side of the breast-butterfly. These blood stains, which are acceptable from a safety viewpoint, could trigger false detection. The potential problems presented by noisy blood stains can be seen in the color patterns in RGB space in Figs. 13(a) and (b), particularly in the image captured by the single-chip camera. As compared in Table III

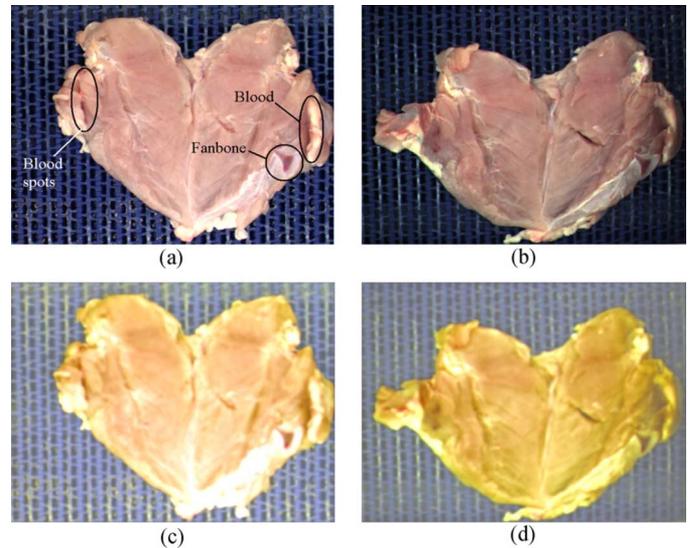


Fig. 12. Example illustrating the effect of sensor resolution. (a) Fanbone (3-CCD). (b) Fanbone (single-CCD). (c) ACC image (3-CCD). (d) ACC image (single-CCD).

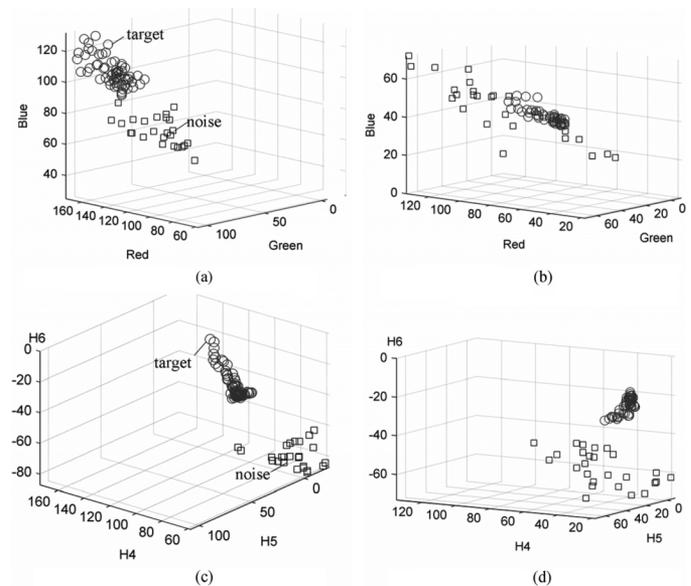


Fig. 13. Comparison of ACC on color features with different image resolution. (a) Color in RGB space (3-CCD). (b) Color in RGB space (1-CCD). (c) Color in ACC space (3-CCD). (d) Color in ACC space (1-CCD).

and Fig. 13, the ACC can effectively reduce the pixel distribution of the fan bone and blood stains in color space. It also increases the separation between the two color clusters. It is worth noting that the preprocessed image of the single-chip camera with ACC could outperform the 3-CCD image without the ACC, implying that the ACC could present a potentially low-cost solution to color classification problems.

Figs. 14(a) and 15(a) show two other example problems commonly encountered in the inspection of food products: grapefruit inspection [34] and detection of contamination in a packaged food product, where MV is used to sort this product based on user-generated parameters. Figs. 14(b) and 15(b) show the grayscale images of the same two products; clearly, color is necessary in order to differentiate between defective areas (such as the blush or contamination) as they would be confused with

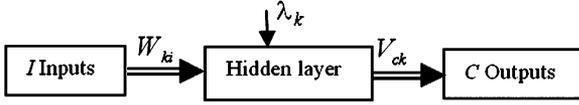


Fig. 16. Structure of the trained network.

TABLE III
EFFECT OF SENSOR RESOLUTION (TARGET)/(NOISE)

	3-CCD image		Single-CCD image	
	Fig. 10(b)	Fig. 10(d)	Fig. 11(b)	Fig. 11(d)
Mean	[150, 92, 108]/	[88, 30, -24]/	[67, 39, 44]/	[36, 9, -24]/
	[138, 62, 67]	[70, -5, -77]	[93, 45, 45]	[50, 7, 57]
SD	[4.1, 6.6, 11.9]/	[1.9, 4.1, 10.9]/	[1.2, 2.3, 9.2]/	[0.6, 2.0, 5.9]/
	[5.3, 8.0, 14.0]	[4.4, 7.1, 11.9]	[5.6, 9.9, 31.2]	[5.1, 7.8, 18.4]
Distance	52	67	27	36

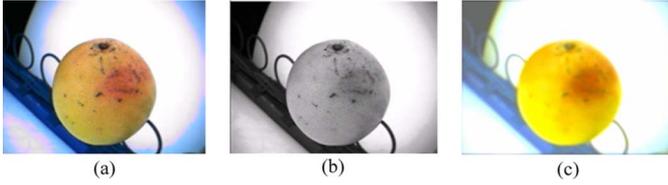


Fig. 14. Grapefruit inspection. (a) Original. (b) Grayscale. (c) Transformed.

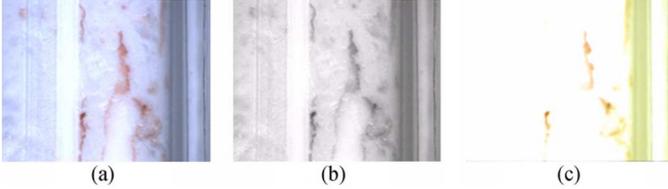


Fig. 15. Detection of contaminations in packaged food. (a) Original. (b) Grayscale. (c) Transformed.

shading. Figs. 14(c) and 15(c) show the transformed images (that are a more desired representation to machines), where color contrast between features is artificially created while noise is suppressed.

IV. EFFECTS OF COLOR CHARACTERIZATION METHODS

We examine the effects of the target-subspace boundary for characterizing the target color by comparing SFBB against two other methods; RCE neural-network classifier (RCE–NNC) and support vector machine (SVM).

A. RCE Neural Network

A three-layer RCE–NNC (shown in Fig. 16) is used to provide the supervised learning of color pattern categories separated by nonlinear, essentially arbitrary boundaries. The concept of a pattern class develops from storing in memory a limited number of class elements (prototypes). Associated with each prototype is a modifiable scalar weighting factor (λ) that defines the threshold for categorizing an input to the prototype. Learning involves 1) commitment of the prototypes to memory and 2) adjustment of the various λ factors to eliminate classification errors.

The three-layer RCE–NNC has I input nodes (equal to the dimension of input vector), C output nodes for the number of output categories, and a hidden layer. The hidden layer is initially empty and creates nodes dynamically through learning. If the new pattern does not belong to an existing class (or is

not within the sphere defined by λ_k), a new node is created in the hidden layer. A default threshold λ and a distance function $D(\mathbf{v}_i, \mathbf{v}_j)$ where \mathbf{v}_i and \mathbf{v}_j are two vectors in the color space must be assigned before training can begin, which could significantly influence the number of nodes generated in the hidden layer and the type of hidden node (for example, sphere or rectangular box), respectively. Thus, these design parameters determine the topology of the trained network and, hence, the performance of the classifier. In this study, the distance function is a 3-D Euclidean distance between two vectors in the color space computed by

$$D(\mathbf{W}_k, \mathbf{R}_n) = \sqrt{(r_k - r_n)^2 + (g_k - g_n)^2 + (b_k - b_n)^2}. \quad (14)$$

The training process of an RCE–NNC is iterative. We illustrate it here using the following pseudocode.

Initialization

Number of nodes on the hidden layer $M = 0$.

Training begins

Let the first pattern \mathbf{R}_1 belong to the c th class resulting in the first hidden node $M = 1$

$$W_{1i} = R_{1i}; V_{c1} = 1; \text{ and } \lambda_1 = \text{default threshold, where } i = r, g, \text{ and } b.$$

The process is repeated and new hidden cells are created.

Confusion

With M cells available, the following process repeats for each new training pattern. Consider the n th training pattern (belonging to the d th class) arrives.

For $k = 1 : M$

$$m_k = N_{\lambda k}(D(\mathbf{W}_k, \mathbf{R}_n))$$

where $D(\mathbf{W}_k, \mathbf{R}_n) = \|\mathbf{W}_k - \mathbf{R}_n\|$ and

$$N_{\lambda k}(x) = \begin{cases} 1, & x < \lambda_k \\ 0, & x \geq \lambda_k \end{cases}.$$

If $m_k = 1$ and $V_{dk} = 1$.

The pattern is correctly contained within a cell.

Else if $m_k = 1$ and $V_{ck} = 1$.

The classification is incorrect. Decrease λ_k until $m_k = 0$.

End

If $\forall k, m_k = 0$, then add a new cell in the hidden layer

$$\mathbf{W}_{M+1} = \mathbf{R}_n$$

$$V_{d(M+1)} = 1, \text{ and}$$

$$M = M + 1.$$

End

End

Once it is trained, RCE–NNC stores the points in a metric space R^N . The boundary of the class is approximated by a set

of hyperspheres in feature metric space. The distance function relates the unknown pattern to a category. The weight of the hidden node is the coordinate of the center of the hypersphere. The threshold of hidden node represents the radius of the hypersphere. A vector in metric feature space will be recognized as one class if it falls into one of the hyperspheres that belongs to that class.

B. Support Vector Machine (SVM)

SVM, a classification method developed from the linear statistical classifier, is the combination of a hyperplane classifier [31] and kernel method [32] [33] and is briefly introduced here for completeness. Given the training set

$$S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\}$$

where $\mathbf{x}_i \in X \subset \mathbb{R}^N$ and $y_i \in Y = \{1, \dots, k\}$ are the predictive (or independent) variable and the target (or dependent variable), we wish to obtain a mapping

$$f_S : X \rightarrow Y.$$

The SVM discussed here is a two-class classification problem, the classes being P, N for $y_i = \pm 1$, which can easily be extended to k class classification by constructing k two-class classifiers.

- *Maximal Margin Hyper-Planes*

If the training data are linearly separable, then there exists a pair (\mathbf{w}, b) such that

$$\mathbf{w}^T \mathbf{x}_i + b \begin{cases} \geq +1, & \text{for all } \mathbf{x}_i \in P \\ \leq -1, & \text{for all } \mathbf{x}_i \in N \end{cases} \quad (15)$$

The decision rule is given by

$$f_{\mathbf{w}, b}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b) \quad (16)$$

where \mathbf{w} is the weight vector; and b is the threshold. As shown in the appendix, the optimal linear hyperplane is constructed in the feature space by applying the margin principle which will maximize the margin between two classes. The decision function is then given by

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^l y_i \lambda_i^* \mathbf{x}^T \mathbf{x}_i + b^* \right) \quad (17)$$

where $\mathbf{w}^* = \sum_{i=1}^l \lambda_i^* y_i \mathbf{x}_i$; $b^* = y_i - \mathbf{w}^{*T} \mathbf{x}_i$; and $\Lambda = (\lambda_1, \dots, \lambda_l)^T$ are the Lagrange multipliers to be obtained by solving the following dual problem given by (A.6) in the Appendix

$$\text{Maximize } F(\Lambda) = \Lambda^T I - \frac{1}{2} \Lambda^T D \Lambda \quad (18)$$

subject to $\Lambda \geq 0$, $\Lambda^T y = 0$; D is a symmetric $l \times l$ matrix with elements

$$D_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j. \quad (19)$$

Note that Λ is only non-zero when $y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$, vectors for which they are called support vectors since they lie closest to the separating hyperplane.

The solution obtained is often sparse since only \mathbf{x}_i with non-zero Lagrange multipliers appears in the solution. This

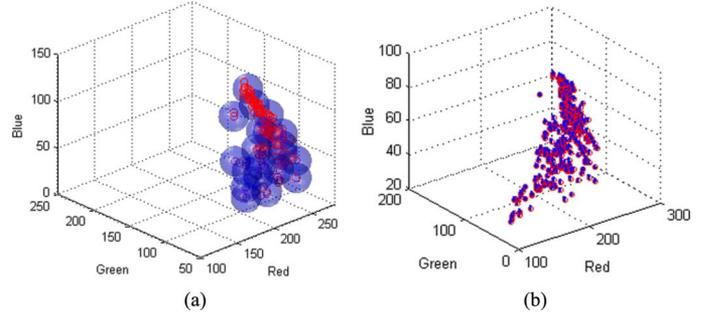


Fig. 17. Computed feature representation in color component space. (a) Trained RCE neural network. (b) Trained SVM.

is important when the data to be classified are very large, as is often the case in practical classification situations. However, it is possible that the expansion includes a large proportion of training data, which leads to a model that is expensive both to store and to evaluate; alleviating this problem is an area of ongoing research in SVMs.

- *Kernel Feature Spaces*

A linear classifier may not be the most suitable hypothesis for the two classes. SVM can be used to learn nonlinear decision functions by mapping the data to some higher dimensional feature space and constructing a separating hyperplane in this space. Denoting the mapping by

$$X \rightarrow H \quad \text{and} \quad \mathbf{x} \mapsto \phi(\mathbf{x}). \quad (20)$$

Mapping data to H is, however, time consuming and storing it may be impossible. Note that H could be infinite dimensional. Hence, a kernel function

$$K(\mathbf{x}, \mathbf{z}) \equiv \phi(\mathbf{x})^T \phi(\mathbf{z}) \quad (21)$$

is introduced and the decision function becomes

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^l y_i \lambda_i^* K(\mathbf{x}, \mathbf{x}_i) + b^* \right) \quad (22)$$

where the bias for any support vector \mathbf{x}_i is given by

$$b^* = y_i - \sum_{j=1}^l y_j \lambda_j^* K(\mathbf{x}_j, \mathbf{x}_i). \quad (23)$$

The kernel function allows us to construct an optimal separating hyperplane in the space H without explicitly performing calculations in this space, which requires K to be an easily computable function; commonly used kernels include linear, polynomial, and radial kernels. In this study, we choose a radial basis function for the kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad \gamma > 0. \quad (24)$$

SVM is reduced to solving the Lagrange multipliers from (16) with $D_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, the solution of which can be obtained using quadratic programming techniques.

C. Comparison of Results

We experimentally examine the effects of the boundary approximation on computational efficiency in color space by comparing SFBB against RCE-NNC and SVM. The studies

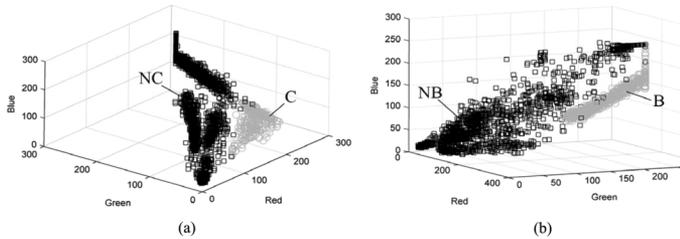


Fig. 18. Training data of SVM. (a) C and NC. (b) B and NB.

were performed using the live-bird orientation detection application as described in Section II-D. Eighty-one (640×480 -pixel) images of grasped birds were captured; and divided into two groups, with 51 birds facing forward and the remaining 30 facing backward. The training sets are the same as in Fig. 4. Thirteen hidden cells are used in REC–NNC. The number of support vectors is 1541. A total of 2401 data was trained. The images were preprocessed with the *DoG* filter to enhance contrast, and postprocessed with a morphological operation based on the majority rule of ten pixels (in an 8×8 mask) to remove isolated noise.

The following summarizes the differences among the three methods and observations made from the results.

- 1) SFBB uses a bounded rectangular box (Fig. 4) with its orientation and dimensions defined by the training set statistics, while RCE–NNC and SVM use hyperspheres and hyperplanes, respectively. To offer a visual comparison, Fig. 17(a) shows the geometry topology of a trained RCE neural network where we choose $\lambda = 8$, minimum threshold = 1, and the distance function defined in (14). The geometry of the generated nodes of an REC–NNC in the hidden layer is spherical in 3-D space. Fig. 17(b) shows the boundary of the same training set but using the SVM classifier. Due to the difficulty in solving the boundary of SVM in closed form, the boundary was numerically computed by testing each pixel cell in RGB space using the SVM classifier. The connected cells are combined to form the boundary. The boundary so generated is not smooth; however, it is intuitive in understanding the SVM classifier boundary.
- 2) While SVM is a more general method for solving classification problems, it has three major disadvantages:
 - 1) Unlike SFBB and RCE–NNC, which can construct the boundary for a single class, SVM needs at least two classes to construct a hyperplane. Although the primary target features are the red comb and white body (as background), there are points in the image that are neither body nor comb. Thus, four classes are needed; (B/NB)—feature color for body and not body, and (C/NC)—for comb and not comb for this problem as shown in Fig. 18.
 - 2) The performance is very sensitive to the training data. The training patterns for nontarget features must be carefully chosen in the proximity of the target in color space so that the boundary of the color subspace is tight.
- 3) The classification results are summarized in Tables IV and V, where the success rates of three methods are compared without and with an ACC prefilter, respectively. The ACC prefilter has helped separate noise from the feature color, which allows the three algorithms to subsequently exclude some of the noise.

TABLE IV
CLASSIFICATION RESULT (WITHOUT ACC PREFILTER)

	Forward	Backward	All	% S Rate
SFBB	50/51	16/30	66/81	81%
RCE–NNC	44/51	15/30	59/81	73%
SVM	30/51	27/30	57/81	70%

TABLE V
CLASSIFICATION RESULT (WITH ACC PREFILTER)

	Forward	Backward	All	% S Rate	Time
SFBB	50/51	30/30	80/81	99%	T
RCE–NNC	51/51	18/30	69/81	85%	1.9T
SVM	30/51	30/30	60/81	74%	13T

where $T = 5.8$ s/image (MATLAB) and $T = 0.5$ s/image (C++)

As shown in Table V, SFBB correctly identifies 80 of 81 cases. The failed case corresponds to an image of a female bird with a small/pale comb, which was captured slightly off timing and, thus, in dim illumination. Written in C++ code, the average cycle time is about 0.5 s/image. RCE–NNC performed poorly in detecting backward facing birds as it introduces excessive noise. On the other hand, SVM missed a number of red combs as the boundary of the color subspace is rather tightly fitted. SFBB appears to have the best potential to meet both cycle time and reliability requirements. In addition, the boundary approximation of an SFBB is relatively straightforward and relies only on the standard deviation of the training data distribution to specify its bounds.

V. CONCLUSION

A new algorithm consisting of a prefilter and a statistically bounded box has been introduced for characterizing the feature color of natural objects for food processing automation. The prefilter artificially creates contrast between features, thereby increasing the distance between clusters of feature pixels in color space and minimizes the appearance of unrelated features as noise. We also show how Hering’s theory of color opponent can be quantitatively applied to the design of color vision algorithms. Despite the emphasis in this paper on food processing applications where red color has been a common dominant color, the techniques for creating an artificial contrast between target and noise features in color space are equally applicable to other processes, such as automation of color-based human face identification or applications with target features other than red.

The statistically bounded box utilizes the principal component analysis technique to characterize target features in color space from a set of training data so that feature color can be more accurately presented and efficiently processed for feature detection. We also examine experimentally the effects of the boundary approximation of the color characterization on computational efficiency. Compared against two other commonly used methods (RCE–NNC and SVM), this algorithm has several advantages including simplicity in training and fast classification since only three simple checks of rectangular bounds are performed. The computational efficiency and reliability of all three methods on color classification have also been evaluated in the context of an automation problem. This study shows that the statistically based fast bounded box can satisfy the stringent requirements of live bid handling automation.

APPENDIX
OPTIMAL LINEAR HYPERPLANE

The inequality constraints (4) can be combined to give

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \text{ for all } \mathbf{x}_i \in P \cup N. \quad (\text{A.1})$$

The pair (\mathbf{w}, b) can be rescaled such that $\min_{i=1, \dots, l} |\mathbf{w}^T \mathbf{x}_i + b| = 1$. The learning problem is, hence, reformulated as minimizing $\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w}$ subject to the constraints of linear separability (A.1). This is equivalent to maximizing the distance (or the normal to the hyperplane) between the convex hulls of the two classes; this distance is called the margin. The optimization is now a convex quadratic-programming (QP) problem

$$\begin{aligned} \text{Minimize } \Phi(\mathbf{w}) &= \frac{1}{2} \|\mathbf{w}\|^2, \\ \text{subject to } y_i(\mathbf{w}^T \mathbf{x}_i + b) &\geq 1, \quad i = 1, \dots, l. \end{aligned}$$

This problem has a global optimum and, thus, avoids the problem of many local optima in NN training. The Lagrangian for this problem is

$$L(\mathbf{w}, b, \Lambda) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \lambda_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] \quad (\text{A.2})$$

where $\Lambda = (\lambda_1, \dots, \lambda_l)^T$ are the Lagrange multipliers, one for each data point. The solution to this QP problem is given by maximizing L with respect to $\Lambda \geq 0$ and minimizing with respect to \mathbf{w} , b . Differentiating with respect to \mathbf{w} and b and setting the derivatives equal to 0 yields

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \Lambda)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^l \lambda_i y_i \mathbf{x}_i = 0 \quad \text{and} \\ \frac{\partial L(\mathbf{w}, b, \Lambda)}{\partial b} &= - \sum_{i=1}^l \lambda_i y_i = 0. \end{aligned} \quad (\text{A.3})$$

So that the optimal solution is given by (6) with a weight vector

$$\mathbf{w}^* = \sum_{i=1}^l \lambda_i^* y_i \mathbf{x}_i. \quad (\text{A.4})$$

Substituting (A.3) and (A.4) into (A.2), we can write

$$F(\Lambda) = \sum_{i=1}^l \lambda_i - \frac{1}{2} \|\mathbf{w}\|^2 = \sum_{i=1}^l \lambda_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (\text{A.5})$$

which can be written in matrix notation

$$F(\Lambda) = \Lambda^T I - \frac{1}{2} \Lambda^T D \Lambda \quad (\text{A.6})$$

where D is a symmetric $l \times l$ matrix with elements $D_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$.

ACKNOWLEDGMENT

The authors would like to thank Prof. B. Webster of Poultry Science at the University of Georgia (UGA) for help with the testing of live birds at the UGA and to Doug Britton and the staff

of the Food Processing Division for help in vision hardware. The birds used in these experiments were provided by Gold Kist Inc.

REFERENCES

- [1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Maui, HI, 1991.
- [2] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. CVPR*, Seattle, WA, 1994.
- [3] X. Mu, M. Artiklar, M. H. Hassoun, and P. Watta, "An RCE-based associative memory with application to human face recognition," in *Proc. Int. Joint Conf. Neural Networks*, Portland, OR, 2003.
- [4] H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *CVPR'98*, Santa Barbara, USA, 1998.
- [5] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 551–564, Jun. 1999.
- [6] N. Habili, L. C. Chew, and A. Moini, "Segmentation of the face and hands in sign language video sequences using color and motion cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 8, pp. 1086–1097, Aug. 2004.
- [7] F. Hsin-Chia, P. S. Lai, R. S. Lou, and H. T. Pao, "Face detection and eye localization by neural network based color segmentation," presented at the IEEE Signal Process. Soc. Workshop, Sydney, Australia, 2000.
- [8] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," *Proc. IEEE*, vol. 83, no. 5, pp. 705–741, May 1995.
- [9] K. M. Lee, W. Daley, and Q. Li, "Artificial color contrast for machine vision and its effects on feature detection," in *Proc. AIM*, Monterey, CA, Jul. 24–28, 2005.
- [10] D. K. Reilly, L. N. Cooper, and C. Elbaum, "A neural model for category," *Learning Biol. Cybern.*, vol. 45, pp. 35–41, 1982.
- [11] T. Fang, W. Fang, P. K. Willett, K. R. Pattipati, and E. H. Jordan, "Signal processing and neural network toolbox and its application to failure diagnosis and prognosis," in *Proc. SPIE*, 2001, vol. 4389, p. 121.
- [12] S. Keyvan, "Traditional signal pattern recognition versus artificial neural networks for nuclear plant diagnostics," *Progr. Nucl. Energy*, vol. 39, p. 1, 2001.
- [13] T. Olmez and Z. Dokur, "Application of InP neural network to ECG beat classification," *Neural Comput. Appl.*, vol. 11, p. 144, 2003.
- [14] S.-M. Roan, C.-C. Chiang, and H.-C. Fu, *Fuzzy RCE Neural Network*. New York: IEEE, 1993.
- [15] M. L. Yuan and M. Xie, "An incremental representation of conceptual symbols using RCE neural network," presented at the 2nd Int. Conf. Development and Learning, Cambridge, MA, 2002.
- [16] Y. Xiaoming, L. Jiangzhou, and X. Ming, "Vision-based human-vehicle interaction and skill training," presented at the 10th Int. Conf. on Advanced Robotics, Budapest, Hungary, 2001.
- [17] K.-M. Lee, J. Joni, and X. Yin, "Imaging and motion prediction for an automated live-bird transfer process," presented at the ASME IMECE DSC, Orlando, FL, Nov. 5–10, 2000.
- [18] M. J. Hudak, "RCE networks: an experimental investigation," presented at the IJCNN, Seattle, WA, 1991.
- [19] T. Olmez, E. Yazgan, and O. K. Ersoy, "Modified restricted Coulomb energy neural network," *Electron. Lett.*, vol. 29, p. 1963, 1993.
- [20] C. Campbell, "Kernel methods: a survey of current techniques," *Neurocomput.*, vol. 48, p. 63, 2002.
- [21] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, p. 121, 1998.
- [22] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT, 1999.
- [23] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," presented at the CVPR, San Juan, Puerto Rico, 1997.
- [24] S. W. Kuffler, "Discharge patterns and functional organization of mammalian retina," *J. Neurophysiol.*, vol. 16, pp. 37–68, 1953.
- [25] P. Lennie, "Parallel visual pathways: a review," *Vis. Res.*, vol. 20, pp. 561–594.
- [26] C. Enroth-Cugell and J. Robson, "Functional characteristics and diversity of cat retinal ganglion cells," *Investigat. Ophthal. Vis. Sci.*, vol. 25, pp. 250–267, 1984.
- [27] E. Hering, *Outlines of a Theory of Light Sense*. Cambridge, MA: Harvard Univ. Press, 1964, (translated by Hurvich and Jameson), (org. 1920).

- [28] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.
- [29] H. Helson, "Fundamental principles in color vision, i—the principle governing changes in hue, saturation, and lightness of non-selective samples in chromatic illumination," *J. Exper. Psych.*, vol. 23, pp. 439–471, 1938.
- [30] D. H. Brainard and B. A. Wandell, "Asymmetric color matching: how color appearance depends on the illuminant," *J. Opt. Soc. Amer.*, vol. 9, pp. 1433–1448, 1992.
- [31] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Autom. Remote Control*, vol. 24, 1963.
- [32] M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical foundations of the potential method in pattern recognition learning," *J. ACM*, vol. 44, no. 4, pp. 615–631, 1997.
- [33] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," presented at the 5th Annu. ACM Workshop Computational Learning Theory, Pittsburgh, PA, 1992.
- [34] D. F. Britton, W. D. Daley, and B. Galloway, "Vision-based citrus inspection and grading system," presented at the Global Int. Signal Processing Conf., Dallas, TX, Mar. 2003.



Kok-Meng Lee (M'89–SM'02–F'05) received the B.S. degree from the State University of New York, Buffalo, in 1980, and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1982 and 1985, respectively.

Currently, he is a Professor with the Woodruff School of Mechanical Engineering at the Georgia Institute of Technology, Atlanta. His research interests include system dynamics/control, robotics, automation, and mechatronics. He holds eight patents in machine vision, three degrees of freedom

(DOF) spherical motor/encoder, and a live-bird handling system.

Dr. Lee is a Fellow of ASME. He has received the National Science Foundation (NSF) Presidential Young Investigator, Sigma Xi Junior Faculty Research, International Hall of Fame New Technology, and Kayamori Best Paper awards.



Qiang Li received the B.S. and S.M. degrees in mechanical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1994 and 1997, respectively, and is currently pursuing the Ph.D. degree at the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta. His research interests include computational methods and modeling, machine vision, control of dynamic systems, and mechatronics.

Mr. Li is the recipient of the AIM2005 Best Student Paper Award.



Wayne Daley received the Ph.D. degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, in 2004.

Currently, he is a Senior Research Engineer with the Georgia Tech Research Institute (GTRI), Atlanta, where he has been for 20 years, focusing on a range of systems that span the implementation of alternative energy systems to machine-vision systems for device control. He is Associate Director of the Food Processing Division with GTRI, where he oversees projects that revolve around the development of next-

generation systems and as well as the development of new programs. His research interests include visual sensing, and the understanding and utilization of human models for artificial vision as applied to MV problems.