IMPROVED IMAGE SEGMENTATION WITH A MODIFIED BAYESIAN CLASSIFIER

Thomas P. Weldon

Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC, USA

ABSTRACT

A method for improving texture segmentation results by slightly modifying the decision surfaces of a Bayesian classifier is presented. Although a Bayesian classifier provides optimum classification within homogeneous regions, it does not necessarily provide accurate localization of region boundaries. In the proposed method, a modified classifier is formed by using a mixture probability density. This approach has the advantage that it is easily implemented in multidimensional classifiers such as those used in classifying the vector output of a filter bank. Experimental results demonstrate improved texture segmentation using the proposed classifier.

1. INTRODUCTION

In practice, a Bayesian classifier based directly on predicted filter bank output statistics performs well within regions, but the locations of boundaries are frequently displaced from their true locations [1-3]. To reduce these errors, a modified Bayesian classifier is proposed that uses a mixture probability density to slightly modify the decision surfaces. Although a modified decision surface could degrade classifier performance within regions, experiments suggest that the improved performance near boundaries outweighs any degraded performance within regions. These problems become even more complex when using multi-resolution filters and feature vectors.

The contradictory demands of low classification error within regions and accurate boundary localization are further complicated when the problem of designing the filters is included in the texture-segmentation problem. While low bandwidth filters tend to improve classification error by reducing feature variance, wider bandwidth filters offer better accuracy near region boundaries. The solution of this larger problem would require joint optimal design of a classifier in conjunction with a bank of filters, and is beyond the scope of the present discussion. Therefore, this paper focuses the smaller problem of improving the segmentation results for a given a set of filters by improving the classifier design. For the purpose of illustrating the proposed method, experimental results are presented for a texture-segmentation example.

One reason for choosing a texture segmentation application is that the feature space is often multidimensional, and is often created with some sort of filter bank. The resulting multidimensional feature space then allows testing of the proposed classifier for the more difficult case of multivariate data. In the texture-segmentation problem, the use of a vector Bayesian classifier offers advantages over previous efforts. Thresholding methods used by Bovik [4] and by Dunn and Higgins [5] imply a per-channel decision scheme that does not take advantage of the multivariate statistics of multiple channels. In other approaches using non-parametric classifiers, Jain and Farrokhnia [6] used a clustering algorithm for classification, and Randen and Husoy [7] used a Kohonen learning vector quantizer. Although such non-parametric classifiers can provide effective classification, they offer limited theoretical insight and may conceal shortcomings elsewhere in the segmentation system.

Previous researchers have also investigated the development of optimal edge detectors in terms of a combined error measure including edge detection and edge localization [8-10]. Bovik also briefly treated the localization problem for a simple scalar decision process [4], but did not combine a classification error with the localization error in his analysis.

Therefore, an image segmentation method is proposed that is based on a classical Bayesian classifier, but with decision surfaces modified to more accurately locate boundaries between regions. The proposed use of a mixture density to modify decision surfaces provides a simple method for improving localization performance of the classifier. In particular, the mixture density approach allows a straightforward implementation for multidimensional feature vectors.

2. APPROACH

To illustrate the new method, a one dimensional classifier is first considered. Then, using insights drawn from the one-dimensional example, the use of a mixture density is proposed to improve the classifier performance at texture boundaries. The mixture density is used to alter the balance between localization errors near region boundaries and classification error within homogeneous regions. After

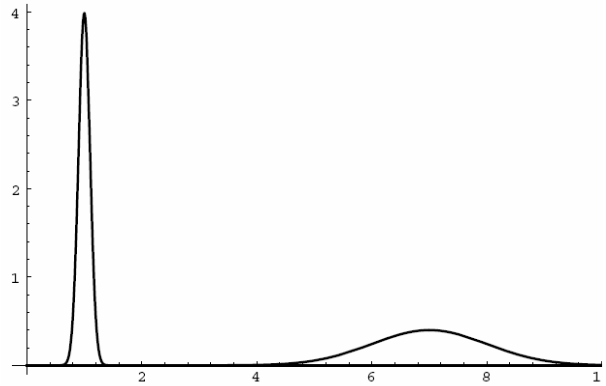

Fig. 1. Two Gaussian density functions corresponding to filter output features for two different classes: $\mu_1 = 1$, $\sigma_1 = 0.1$; $\mu_2 = 7$, $\sigma_2 = 1$, optimum classification threshold = 1.6.

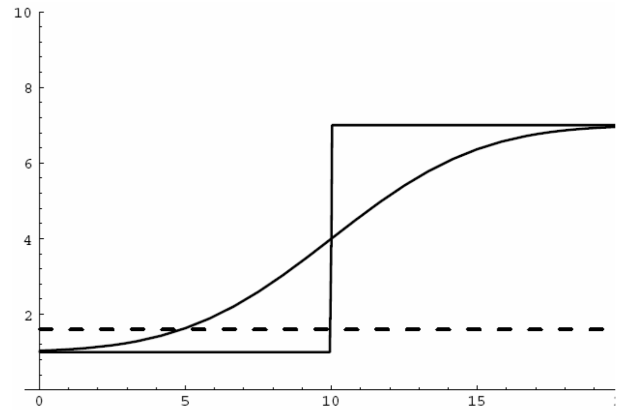

Fig. 2. One-dimensional step corresponding to texture boundary, with second solid curve corresponding to filtered step, and dashed line corresponding to optimum threshold from Fig. 1. The threshold of 1.6 applied to the filtered step results in a boundary displacement error of 5 units to the left of the actual boundary location.

the one dimensional example is discussed, the mixture density is extended to the case of a multivariate classifier. The simplicity of using the mixture density notion for the more difficult multivariate problem is the primary motivation for choosing this approach.

To illustrate the mixture density's effect on classifier performance, it is useful to first consider the two one-dimensional Gaussian density functions depicted in Fig. 1. The two probability density functions in Fig. 1 represent the output statistics of a single feature vector (i.e., the output of a single filter), with each function corresponding to a different class or image region. The optimum classification threshold of 1.6 coincides with the intersection of the two density functions, assuming the two classes are equally likely. If this threshold were used, the classifier would minimize error within homogeneous regions of either class. Note that a second optimal threshold exists, but does not contribute materially to the error for the case shown.

Next, consider the optimum threshold for minimizing the error due to incorrectly locating the position of a boundary between two regions. A one-dimensional step from a feature amplitude of 1 to an amplitude of 7 is shown in Fig. 2. The values of 1 and 7 on either side of the step correspond to the mean values of the density functions in Fig. 1 and, therefore, represent a transition between the two classes corresponding to the two density functions. Filtering this step with a Gaussian lowpass filter results in the second solid curve and corresponds to the effect of spatially filtering the boundary, as commonly done in creating features for segmentation.

The optimal threshold for the filtered step of Fig. 2, from standpoint of accurately locating the boundary, is at the intersection of the two solid curves in Fig. 2. This point is at the mean of the two amplitudes $(\mu 1 + \mu 2)/2 = (7+1)/2 = 4$. The optimum classification threshold of 1.6 is shown as a dashed line in Fig. 2; applying this threshold to the filtered step would result in a displacement error of 5 units to the left of the step in locating the step boundary. Thus, the optimum classification threshold of 1.6 conflicts with the optimum localization threshold of 4.

This is the commonly encountered filtering tradeoff between the reduction of noise within regions and the blurring of boundaries between regions. However, it is also possible to modify the classifier to tradeoff increased error within regions for reduced error at boundaries between regions. In the foregoing example, the situation of Fig. 1 represents a well-separated class pair with a low associated classification error. It is apparent that changing the classification threshold from 1.6 to 4 in Fig. 1 will not seriously degrade classification error. In fact, for this particular example the classification error within homogeneous regions would change from a value of approximately 10^{-9} to a value of 10^{-3}. On the other hand, a 5 pixel boundary error as in Fig. 2 would correspond to a boundary localization error of $(5\times4\times100)/256^2 = 0.03$ for a 100×100 pixel square region in a 256×256 image. This implies the potential to advantageously reduce the localization error from 0.03 to some lower value, at the relatively insignificant cost of a classification error of 0.001.

Therefore, by placing the threshold nearer to the localization threshold of 4, localization error is greatly reduced while only modestly increasing classification error. One method to achieve this shift in threshold is to artificially distort the original density functions in Fig. 2 using a mixture density to generate the density functions shown in Fig. 3. It is emphasized that this distortion is performed only to shift the classifier decision thresholds and does not represent any change in the statistics of the filtered features. Below, details of the mixture density are next discussed.

3. MULTIVARIATE CASE

In the multidimensional case, similar displacement of segmentation boundaries occurs in experiments with multivariate classification of the filtered feature vectors. The following discussion illustrates the proposed method for the particular case of multidimensional feature vectors as would be generated in multi-channel filter decompositions in texture segmentation.

Earlier results [1-3] predict a multivariate Gaussian model for the vector output statistics of filtered texture features. In this, the probability density for class $\boldsymbol{i}$ is $p_i(\boldsymbol{m},\mu_i,\boldsymbol{C}_i)$ where

$$p_i(m,\mu_i,C_i) = \frac{1}{(2\pi)^{k/2}\,|C_i|^{1/2}} e^{-(m-m_i)^T C_i^{-1}(m-m_i)/2}$$

where $\boldsymbol{m}$ is the k-dimensional feature vector, μ_i is the k-dimensional feature vector mean for class $\boldsymbol{i}$, $\boldsymbol{C}_i$ is the $k \times k$ covariance matrix for class $\boldsymbol{i}$, and k is the number of features. Given the multivariate statistics, the Bayesian classifier would then classify any measured feature vector $\boldsymbol{m}_p$ as being a member of class α, if

$$p_\alpha(m_p,\mu_i,C_i) > p_\beta(m_p,\mu_i,C_i); \quad \forall\, \beta \in \{1, 2, \ldots, N\},$$

where $\alpha \in \{1, 2, \ldots, N\}$, and N is the number of classes.

However, when a Bayesian classifier based on the predicted multivariate Gaussian density function is employed, experiments show a large amount of localization error at texture boundaries; i.e., the boundary is displaced from its true location.

Based on the preceding one-dimensional argument and based on experimental findings, the classifier is modified using a mixture-density consisting of the average of two multivariate-Gaussian densities. The mixture density provides an effective means for shifting the classifier's multi-dimensional decision surface in a manner that reduces localization error near texture boundaries while not significantly degrading the classification error within homogeneously textured regions. This is readily implemented for the multivariate-Gaussian statistics of the filter-channel outputs.

To construct the proposed classifier, first construct covariance matrix $\boldsymbol{C}_{max}$ by using the the largest elements in the N covariance matrices $\boldsymbol{C}_i$. Then define the mixture density as

$$p_{imix}(m) = \{\, p_i(m,\mu_i,C_i) + p_i(m,\mu_i, C_{max}) \,\}/2.$$

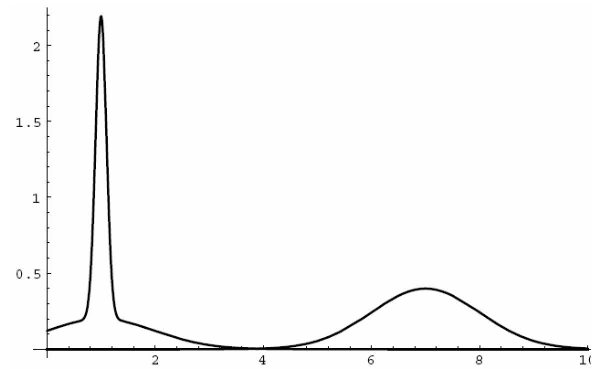

Fig. 3. Two mixture densities based on the original densities of Fig. 1, with new optimum classification threshold of 4 that would greatly reduce the boundary displacement seen in Fig. 2.

The Bayesian classifier with the mixture distribution is then simply implemented for any measured feature vector $\boldsymbol{m}_p$ by classifying it as a member of class α, if

$$p_{\alpha mix}(m_p) > p_{\beta mix}(m_p)\ ; \quad \forall\, \beta \in \{1, 2, \ldots, k\}.$$

Fig. 3 illustrates the mixture densities for case shown in Fig. 2. By inspection of Fig. 3, the new decision threshold is near a value of 4, where the two densities intersect. This new threshold near 4 closely coincides with the optimum localization threshold of 4 seen in Fig. 2.

4. RESULTS

The new method was tested on a wide range of natural and synthetic textured images. In these experiments, the classifiers were implemented with the mixture-density classifier. Experiments employed 256x256 pixel 8-bit gray-scale images where the average grayscale of all textures were equalized to prevent biased segmentation results that may be caused by leakage of the DC component through the filters.

The effects of the mixture-density classifier are shown in Fig. 4. The image in Fig. 4(a) consists of three textures (N=3): an outermost region of noise, a middle ring of d21 "french canvas," and an innermost square region of d55 "straw matting" from the Brodatz texture album. With misclassified pixels in black, Fig. 4(b) shows the segmentation error when the mixture density is not used; i.e., Bayesian classification without a mixture density. For this segmentation, three filters were used (k=3), resulting in a 3-dimensional feature vector $\boldsymbol{m}$. Fig. 4(c) shows the segmentation error when the mixture density is used for classification. The improvement near texture boundaries that is apparent in comparing Figs.4(b) and (c) is confirmed by measurements. Total classification error is 10% in Fig.

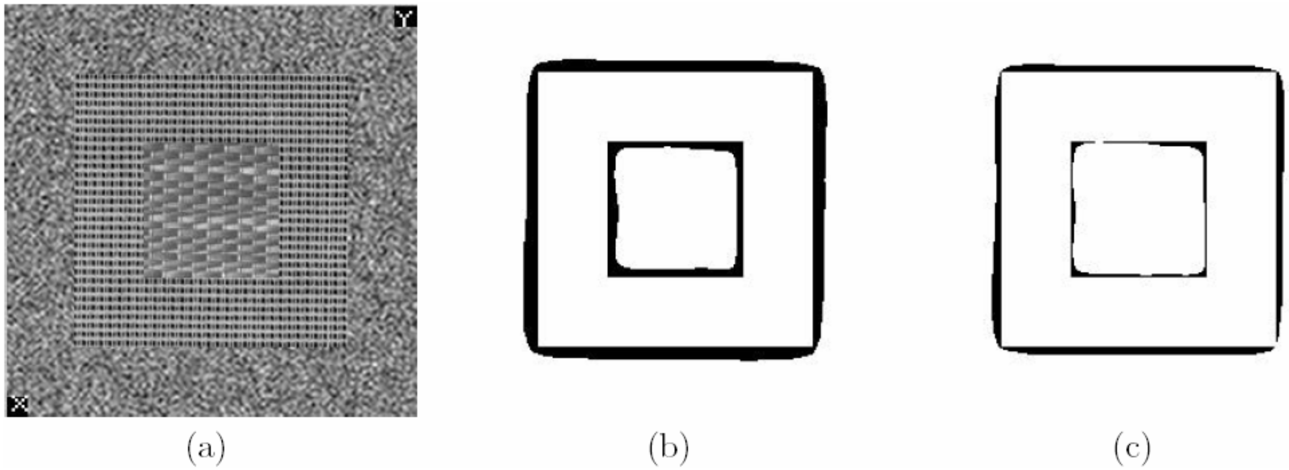

Fig. 4. Effects of mixture density. (a) Input composite image, outer border = uniform noise, middle ring = d21 "french canvas," center square = d55 "straw matting." (b) Segmentation error for (a) without mixture distribution, black = misclassified pixel, measured error=0.10. (c) Segmentation error for (a) with mixture distribution, measured error=0.05.

4(b) without using the proposed mixture density, and is reduced to 5% in Fig. 4(c) when the mixture density is used.

4. SUMMARY

The experimental results demonstrate the effectiveness of the modified Bayesian classifier in reducing total segmentation error. These results show that a mixture density can be used to improve classifier performance near classification boundaries.

5. REFERENCES

[1] T. P. Weldon and W. E. Higgins, ``Designing Multiple Gabor Filters for Multi-Texture Image Segmentation," *Optical Engineering* , Vol. 38 No. 9 , pp. 1478-1489 , Sept. 1999.

[2] T. P. Weldon, W. E. Higgins, and D. F. Dunn, ``Efficient Gabor Filter Design for Texture Segmentation," *Pattern Recognition ,* Vol. 29, No. 12, pp. 2005-2015, Dec. 1996.

[3] T. P. Weldon, W. E. Higgins, and D. F. Dunn, ``Gabor Filter Design for Multiple Texture Segmentation," *Optical Engineering*, Vol. 35 No. 10, pp. 2852-2863, October 1996.

[4] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. Signal Processing*, vol. 39, no. 9, pp. 2025-2043, Sept. 1991.

[5] D. F. Dunn and W. E. Higgins, "Optimal Gabor Filters for texture segmentation," *IEEE Trans. Image Proc*., vol. 4, no. 7, pp. 947-964, July 1995.

[6] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor Filters," *Pattern Recognition*, vol. 23, no. 12, pp. 1167{1186, Dec. 1991.

[7] T. Randen and J. H. Husoy, "Novel approaches to multichannel Filtering for image texture segmentation," in *Proc. SPIE Visual Comm. Image Proc.*, vol. 2094, pp. 626-636, 1994.

[8] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell*., pp. 679-698, June 1986.

[9] H. D. Tagare and R. J. P. deFigueiredo, "On the localization performance measure and optimal edge detection," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 12, pp. 1186-1190, Dec. 1990.

[10] R. Kakarala and A. O. Hero, "On achievable accuracy in edge localization," *IEEE Trans. Pattern Anal. Machine Intell*., vol. 14, no. 7, pp. 777-781, July 1992.

[11] P. Brodatz, Textures: *A Photographic Album for Artists and Designers*. New York, NY: Dover, 1966.

[12] Anil K. Jain, Fundamentals of Digital Image Processing, Englewood Cliffs, NJ, Prentice Hall, 1989.