

# Color Image Segmentation Using Local Histogram and Self-Organization of Kohonen Feature Map

You-Shen Lo and Soo-Chang Pei

*Department of Electrical Engineering  
National Taiwan University  
Taipei, Taiwan, R.O.C.  
E-mail:pei@cc.ee.ntu.edu.tw*

## Abstract

*Segmentation is an important step for image analysis, but a good segment algorithm which can handle color image with texture area and has less computation time is rare. We propose to use the local window image histogram, which is easy to compute and could quickly collect the information of neighbors, together with the Self-Organization of Kohonen Feature Map (SOFM) neural network, which can efficiently cluster data and has parallel hardware structure, as a segmentation kernel. Under the Euclidean distance function with input data normalization and the simplified Mahalanobis distance function, this algorithm will have very good segmentation results for natural images either full with texture or mixed with smooth scenes.*

## 1 Introduction

Segmentation is an important pre-processing step for many inspections or detections in image analysis. Generally speaking, segmentation could be classified into three major classes [1]: (1) characteristic feature thresholding or clustering, (2) edge detection, and (3) region extraction.

The feature thresholding or clustering methods [2, 3] usually project all the image data into the color feature space to find the histogram, then try to threshold the histogram iteratively on the spatial domain and the histogram plane to segment the images [2]. More often, the researchers get used to take the color spaces as the feature spaces [2, 3]. Color feature thresholding approaches assume each object in the image occupies its own color cluster in the color space. On the other hand, pixels with the same color may spread out around the the whole image, purely using thresholding method, sometimes, can not clearly segment out the objects that we want. Actually, images bear more or less a few portions of texture area. Therefore, many researchers try to use some compli-

cated features, like mean, energy, inertia, entropy and contrast etc. to segment images [4]. These features are derived from the spatial gray level dependence matrix (SGLDM). Though these features will involve more spatial information, they also introduce more computational loading.

Self-Organization of Kohonen Feature Map (SOFM) is a very powerful tool for data clustering. It has shown successfully on image quantization [5, 6]. Since segmentation is a process of clustering, therefore, it is also suitable to apply SOFM computation.

In this paper, we use SOFM neural network as a training tool and adopt the index-counts of the local histograms as the clustering features. First, we use SOFM trainer to do color quantization on a true natural color image to get an indexed image. Second, we build index-count vectors of each local histogram from the indexed image. Then, we use the index-count vectors (or features) as the second step of SOFM training data. Finally, after the training, we map back the borders between each cluster from the index-count space to the original image. And we get a well-segmented result.

In section 2, we describe the local histogram based segmentation algorithm. In section 3, we show the simulation results and have discussions on distance functions, numbers of classes and window sizes. And finally, we make some conclusions.

## 2 Local Histogram Based Segmentation Algorithm

The proposed local histogram based segmentation algorithm is a two-step training algorithm. First, we apply SOFM trainer to do the color quantization for input true color images. It is a data reduction process. Too much data and information will cause the computation loading too heavy. And the color quantization is a fast and efficient data reduction process for

true color images. Our previous paper [5] uses SOFM method to do color quantization and gets the indexed images. This method will reduce the redundant information and also keep the borders' information for segmentation.

Second, any image may have its own histogram. The histogram reveals part of characteristics of that image. But dividing the feature spaces into a few clusters by thresholding, still there are some portions of image hardly to find their correct clusters. Due to the vague separation on global histogram, we use the local histograms as our dynamic segment features. The indexed image is a kind of image with fewer color indexes and its color table. Therefore, we can calculate the index histogram of that indexed image. Instead of calculating the global histogram of the index image, we calculated the local histograms at a  $n_w \times n_w$  moving window centered at pixel  $(i, j)$  and sliding over the whole index image. We call one of the entries of the local histogram as an index-count, which is the counting number of one specified index in the moving window at pixel  $(i, j)$  in that index image. Since the pixel  $(i, j)$  is around with  $n_w \times n_w - 1$  neighboring pixels, the local histogram calculation, dealing with  $n_w \times n_w$  window of pixels, will include in all the neighboring (spatial) information. We have three major reasons to support the index-counts as the train features: First, the index-count vectors are omnidirectional, therefore, features are not biased for any direction. We do not have to worry about the direction problem. Second, local index-counts are around pixel  $(i, j)$ , so they could collect local and neighboring information. Third, local index-counts are easily calculated and very low complexity.

Then, we apply SOFM training algorithm to the local histogram index-count vectors of the first step output indexed image. For two pixels, if their index-counts are similar, they should belong to the same class. According to the characteristics of index-counts, the SOFM trainer would classify the indexed image into several pre-defined classes; that is, we use the index-counts of local histograms as the features to classify the image into the predefined clusters. And the borders of the clusters form the segmented pieces of the image. Then we get our final segmented image results. See Figure 1 for the flowchart of the algorithm.

We have tried several distance function for training; such as: the Euclidean distance function, the inner product distance function, the min-max distance function, the Mahalanobis distance function and the simplified Mahalanobis distance function. Each training case may have input data with normalization or

not. The equations are shown as below. The Euclidean distance function:

$$\min_i \text{dist}(\mathbf{x}, \mathbf{m}_i) = \sum_{j=1}^J (x_j - m_{i,j})^2 \quad . \quad (1)$$

The inner product distance function:

$$\max_i \text{dist}(\mathbf{x}, \mathbf{m}_i) = \sum_{j=1}^J x_j \cdot m_{i,j} \quad . \quad (2)$$

The min-max distance function:

$$\max_i \text{dist}(\mathbf{x}, \mathbf{m}_i) = \sum_{j=1}^J \min(x_j, m_{i,j}) \quad . \quad (3)$$

The Mahalanobis distance function:

$$\min_i \text{dist}(\mathbf{x}, \mathbf{m}_i) = (\mathbf{x} - \mathbf{m}_i)^T C_i^{-1} (\mathbf{x} - \mathbf{m}_i) \quad . \quad (4)$$

The simplified Mahalanobis distance function:

$$\min_i \text{dist}(\mathbf{x}, \mathbf{m}_i) = \sum_{j=1}^J \frac{(x_j - m_{i,j})^2}{c_{i,j}} \quad , \quad (5)$$

where  $\mathbf{x}$  is the input local index-count vector,  $\mathbf{m}_i$  is the  $i$ 'th neuron's value and the  $c_{i,j}$  is the each entry of the covariance matrix of all input data.

When the distance measure for training is the Euclidean distance function, we should use the minimum criterion for two-vector difference. If we use inner product distance function as the distance measure, we should use the maximum criterion for two-vector inner product, the same as the min-max distance function. Other distance functions all apply the minimum criterion.

### 3 Simulation Results and Discussions

We test the "lake" color image for segmentation, and the segmented results are on Figure 2. Figure 2 (a) is the segment result for lake whose window size is  $7 \times 7$  and the distance function is Euclidean distance for 4 clustering classes. Also, the input data have been normalized at every training step. We could clearly see from the segmented image of lake that the homogeneous portions (with and without texture) are clearly segmented out. There is no objective evaluation function for segmentation, but we could see from the segmented pictures that the segmented results are fairly good. Hence, our proposed algorithm is very powerful for image segmentation.

When considering the spatial interaction due to the window sizes, we show the segmented result of  $15 \times$

15 window size for comparison. Figure 2(b) is the result with the same condition as Figure 2(a) except for the window size. With larger window size, the segmentation results degenerate in some area and can not match the borders of the objects. Too small the window size may have more detail segment and better alignment but causes more ragged edges.

As for different distance functions, we have tried Euclidean distance function, inner product distance function, min-max distance function, Mahalanobis distance function and simplified Mahalanobis distance function. Each training case may have input data with normalization or not. This may have great influence on training results. The segmented results applied the Euclidean distance function and with input data normalization are better than those cases without normalization.

When increasing the cluster classes from 4 to 8, the segmentation results have more ragged detail area. The darker leaves and lighter leaves are easily separated out. Figure 2(c) shows the result. The more classes the image data could be assigned, the more detailed segments we can see. Hence, if we desire more details of the image, we could increase the number of classes. This is suitable for segmentation-based image compression.

The results which applied simplified Mahalanobis distance function are better than the Mahalanobis distance function, and their results are shown on Figure 2(d).

There is no significant difference when the number of color index is 16 or 32. Therefore, we do not show any result of this part.

The final discussion is about the training features. Compared to many features derived from SGLDM [4], which many papers use [4], we use the index-counts of local histogram as the training features. The SGLDM is a 2-dimensional distribution based on the number of occurrences from one gray level to another. And those features, like mean, variance, correlation, energy, entropy, contrast and homogeneity, are calculated from the SGLDM. The first layer of SGLDM calculation is as simple as our index-count calculation of local histogram. But the second layer of the above features are derived from the complicated calculation and with many multiplications. Even there is a fast algorithm to find those features in the second step [4], it still takes time. Therefore, our training features will be a simple calculation.

## 4 Conclusions

The significant concept of our local index histogram based segmentation algorithm is that the color quan-

tization is the first rough segmenting process on the color space and the index-counts of local histogram training is the second stage fine process for segmentation. The first stage of operation reduces the amount of data for the second stage training. And, at the second stage, instead of using SGLDM [4] parameter as the training feature, we find that the index-counts of local histogram is a nice feature for segmentation. Index-counts are omnidirectional, full of spatial and local information and easy to compute. Hence, our algorithm not only considers the neighboring information of the image for segment, but also greatly reduces the computation burden. Meanwhile, SOFM is a fast training method and has parallel hardware structures. Therefore, it can segment images very fast and well no matter the images are mixed with smooth or texture areas.

## References

- [1] N. R. Pal and S. K. Pal, "A review on image segmentation techniques", *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.
- [2] M. Celenk, "Analysis of color images of natural scenes", *Journal of Electronic Imaging*, vol. 4, no. 4, pp. 382-396, 1995.
- [3] S. Tominaga, "Color classification of natural color images", *Color Research and Application*, vol. 17, no. 4, Aug. 1992.
- [4] Z. Wang, A. Guerriero, M. De Sario, "Comparison of several approaches for the segmentation of texture images", *Pattern Recognition Letters*, vol. 17, pp. 509-521, 1996.
- [5] S. Pei and Y. Lo, "Color image compression and limited display using self-organization Kohonen map", *IEEE Trans. on Circuit and System for Video Technology*, vol. 8, no. 2, pp.191-205, Apr. 1998.
- [6] N. M. Nasrabadi and Y. Feng, "Vector quantization of images based upon Kohonen self-organizing feature maps", *Proc. IEEE ICNN*, pp. I101-I108, 1988.

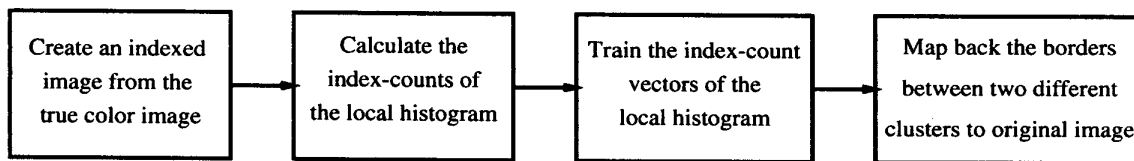


Figure 1: Local histogram based image segmentation algorithm.

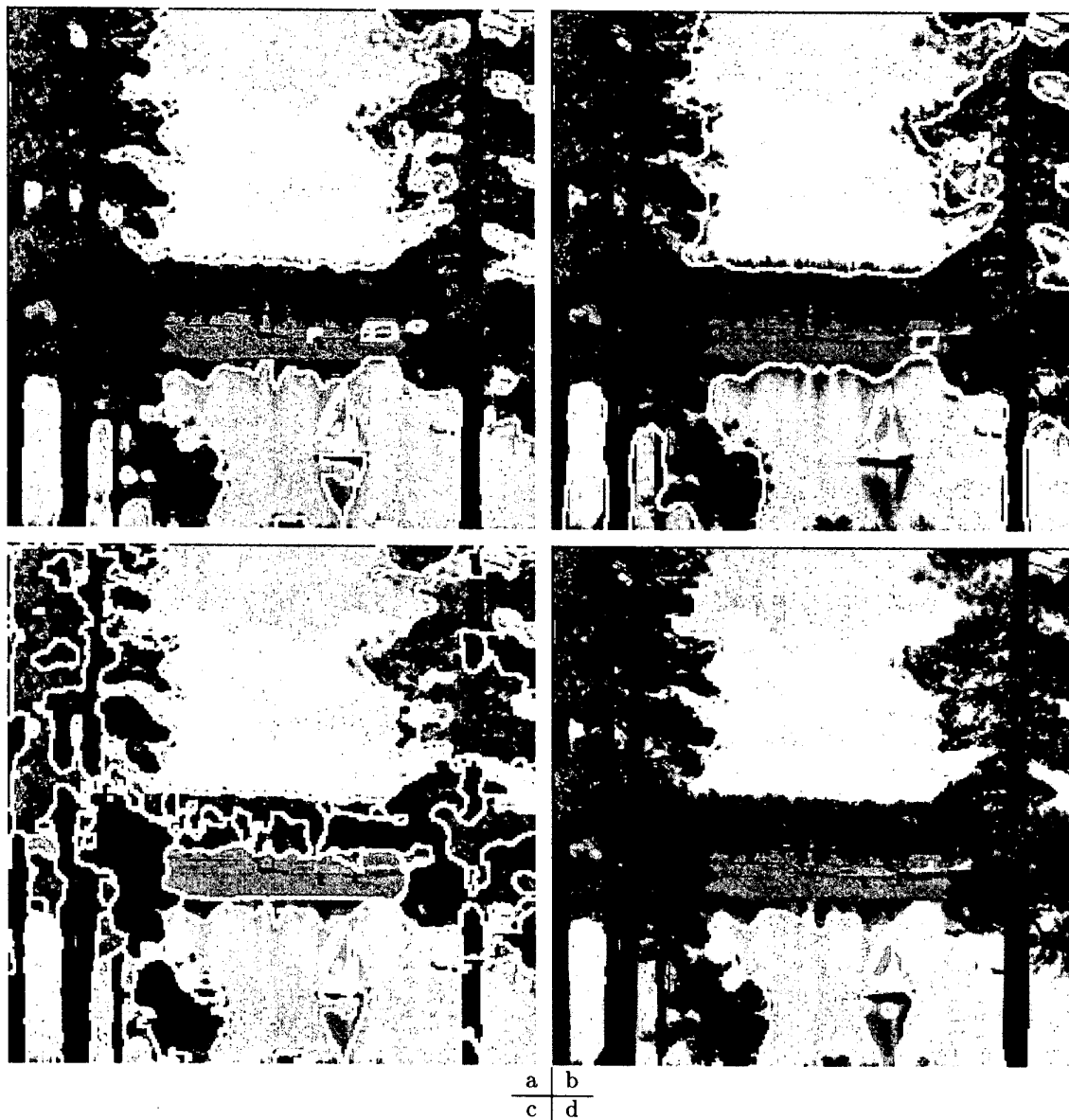


Figure 2: Segmented results : (a) window size  $7 \times 7$  , 4 classes , Euclidean distance function with data normalization. (b) window size  $15 \times 15$  , 4 classes , Euclidean distance function with data normalization. (c) window size  $7 \times 7$  , 8 classes, Euclidean distance function with data normalization. (d) window size  $7 \times 7$  , 4 classes , simplified Mahalanobis distance function.