Segmentation

Using Hopfield Networks to Segment Color Images

P. Campadelli*, D. Medici*, R. Schettini°

*Dip. di Scienze dell'Informazione, Universita' degli Studi di Milano,
Via Comelico 39, Milano, Italy

°Istituto Tecnologie Informatiche Multimediali, Consiglio Nazionale delle Ricerche
Via Ampere 56, 20131 Milano, Italy

Abstract. In this paper we present two algorithms for color image segmentation based on Huang's idea of describing the segmentation problem as the one of minimizing a suitable energy function for a Hopfield network. The first algorithm builds three different networks (one for each color feature) and then combine the results. The second builds a unique network according to the number of clusters obtained by histogram analysis. Experimental results, heuristically and quantitatively evaluated, are encouraging.

1. Introduction

The segmentation of color images is frequently based on supervised pixel classification when prior knowledge about the object colors is available, while clustering and histogram thresholding are the most widely used unsupervised methods. The algorithms employed assume that homogeneous regions correspond to distinct clusters in the feature space without taking into account spatial information, and may, therefore, generate noisy results. Relaxation algorithms have been proposed to modify the initial cluster labels of the segmented image on the basis of their spatial context [4]. In 1992 Huang [3] described the segmentation problem for grey-level images as the problem of minimizing a suitable energy function for Hopfield networks. We present here two algorithms for color image segmentation based on Huang's idea and evaluate their performance quantitatively.

2. Histogram analysis

We can, by applying one-dimensional histogram analysis to each color component, partition the color space into several hexahedra corresponding to prominent clusters in the color space. A segmented image can be obtained by assigning the mean color of each cluster to the corresponding pixels of the image. The method assumes that homogenous regions in the image correspond to color clusters which when projected on the feature axis generate at least a visible histogram peak, and that these clusters lie apart from each other in the feature space (i.e. the number of clusters can be reliably found). Segmentation results depend largely on the methods adopted to detect "significant" peaks and their respective cut-off values in the histograms. Since in image segmentation the number of clusters and their statistical descriptions are usually not known a priori, we can not use statistical methods to obtain optimal decision boundaries for the peaks in the histograms. We adopt, instead, a technique based on scale-space histogram filtering [7].

3. The Hopfield model

Symmetric networks of non linear graded-response neurons [2] are composed of units whose input, output and state assume values within the interval [0,1] (or [-1,1]). The input u_i to neuron i $(1 \le i \le n$, where n is the number of network units) is the weighted sum of the outputs of the units connected with neuron i: $u_i = \sum_j w_{ij} V_j$, where w_{ij} is the weight of the connection between neuron j and neuron i, V_j is the output of neuron j $(1 \le j \le n)$. The input u_i is converted into the output value V_i by a continuous monotone-increasing function g_i: $V_i = g_i(u_i) = g_i(\sum_j w_{ij} V_j)$. When u_i and V_i vary within the range [0,1], a commonly used function is $1/2[1+tanh(\beta\ u_i)]$.

The following set of coupled differential equations describes how the state variable u_i $(1 \le i \le n)$ of the neurons changes with time:

$$\frac{du_i}{dt} = -\frac{u_i}{\tau_i} + \sum_j w_{ij} V_j \qquad (1)$$

where τ_i are suitable time constants.

As Hopfield [2] has shown, the evolution in time of a symmetric network of analog neurons is a motion in the state space that seeks out minima in the function:

$$E_a = -\frac{1}{2}\sum_i \sum_j w_{ij} V_i V_j + \int_0^{V_i} g_i^{-1}(V)dV \qquad (2)$$

When β (the parameter responsible for the steepness of the sigmoid activation function g_i) is large, the second term in expression (2) is negligible, and the network minimizes locally the function :

$$E = -\frac{1}{2}\sum_{i \neq j} w_{ij} V_i V_j \qquad (3)$$

This is the Lyapunov function for networks of binary threshold neurons with a symmetric connection matrix (of non-negative diagonal elements) and random sequential updating [1]. Takefuij [6] has proved that a network of analog neurons can always be made to converge in the local minima of expression (3), when its dynamical behavior is described by the following set of differential equations:

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial V_i} \qquad (4)$$

that is, the term u_i/τ_i in equation (1) has been eliminated.

4. Segmentation algorithms

In 1992 Huang [3] described the segmentation problem for grey-level images as one of minimizing a suitable energy function for Hopfield networks: Let $f(x,y)$ be an

NxM grey-level image to be segmented and S (estimated by histogram analysis) the maximum number of classes to be obtained by the segmentation process. Consider a Hopfield network made of $NxMxS$ analog neurons organized in S layers of NxM neurons each. Each neuron in a layer represents an image pixel and is denoted by the triple (x,y,i), where (x,y) $(1 \leq x \leq N, 1 \leq y \leq M)$ are the pixel coordinates and i $(1 \leq i \leq S)$ denotes one of the S classes. The output $V_i(x,y)$ $(V_i(x,y) \in [0,1])$ of the neuron (x,y,i) is interpreted as the probability that pixel (x,y) will be assigned to class i. The function E, from which the network architecture is derived, expresses the constraints of the segmentation problem and is the sum of three terms: $E = AE_1 + BE_2 + CE_3$, where A, B, and C are suitable, positive constants and E_1, E_2, E_3 have the following expressions:

$$E_1 = \sum_{x=1}^{N} \sum_{y=1}^{M} \sum_{i=1}^{S} \sum_{l=-1}^{1} \sum_{k=-1}^{1} (V_i(x,y) - V_i(x+l, y+k))^2 \tag{5}$$

$$E_2 = -\sum_{x=1}^{N} \sum_{y=1}^{M} \sum_{i=1}^{S} \sum_{J=1, J \neq i}^{S} \sum_{l=-1}^{1} \sum_{k=-1}^{1} (V_i(x,y) - V_j(x+l, y+k))^2 \tag{6}$$

$$E_3 = \sum_{x=1}^{N} \sum_{y=1}^{M} (\sum_{i=1}^{S} V_i(x,y) - 1)^2 \tag{7}$$

Observe that the term E_1 favors the assignment of adjacent pixels to the same class; the term E_2 is minimum when a pixel and its neighbors are assigned to only one class, and, finally, the term E_3 is zero if, and only if, the sum of the outputs of the neurons corresponding to the same pixel is equal to one. In his algorithm Huang initialized all the neurons, corresponding to the pixels belonging to a given peak, at the same values and let the network evolve according to a simulated annealing algorithm.

We have devised two different algorithms, the first one, which resembles Huang's algorithm for grey-level images, builds three different networks (one for each color feature considered) and then combine the results. The second builds a unique network according to the number of clusters obtained by histogram analysis. In both of these algorithms we have changed both the network initialization and its dynamic evolution with respect to the original proposition.

ALGORITHM 1

Input: Color Image

FOR each color component DO:

- Apply histogram analysis (Section 2) to locate significant peaks;
- Build a network with a number of layers equal to the number of peaks;
- Initialize each network neuron (x,y,i) according to the following rule:

$$V_i(x,y) = \frac{1/d_i(x,y)}{\sum_j 1/d_j(x,y)} \tag{8}$$

where $d_i(x,y) = |f(x,y) - p_i|$ and p_i is the location of the i-th peak;

- Let the network evolve according to equation (4) until each neuron output changes less than a predefined threshold T.

END

Combine the segmentation results (union of edges).

Output: Segmented image

ALGORITHM 2

Input: Color Image

- Partition the color space into several hexahedra corresponding to prominent clusters in the color space;
- Calculate the centroids of the selected clusters;
- Build a network with a number of layers equal to the number of selected clusters;
- Calculate for each pixel *(x,y)* the Euclidean distance d_{C_i} between the pixel color and each cluster centroid C_i;
- Initialize each network neuron *(x,y,i)* according to the following rule:

$$V_i(x,y) = \frac{1/d_{C_i}(x,y)}{\sum_j 1/d_{C_j}(x,y)} \tag{9}$$

- Let the network evolve according to equation (4) until each neuron output changes less than a predefined threshold T.

Output: Segmented image

5. Evaluation of the algorithms

We have tested Algorithm 1 and Algorithm 2 on many images in several different domains. In both algorithms, parameters *A, B, C,* and T must be set. We have found experimentally that the choice of the parameters *A* and *B* is not critical (and have set both of them at 0.04). The parameter *C* has been increased by steps of 0.1 during network evolution from an initial value of 1 to a value equal to the number of layers. We have observed that varying C according to this rule makes the network converge more quickly to solutions satisfying equation (7). The threshold T for the termination condition has been set at 0.0000001. In order to evaluate the performance of the algorithms more objectively we have adopted the evaluation function recently proposed by Yang and Liu [5]. This function is defined as

$$F(I) = \frac{1}{1000(N \times M)} \sqrt{R} \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}} \tag{10}$$

where I is the image to be segmented, NxM, the image size, and R, the number of regions of the segmented image, while A_i and e_i are the area and the average color error of the *i-th* region respectively.
As an example, the segmentation of a 256x256 24-bit pixels test image is reported here. The image in figure 1a represents a aircraft photograph digitalized by a flat bed scanner. Figure 1b shows the segmentation obtained by the histogram thresholding of each RGB color feature independently, without taking into account spatial correlations among pixels. Figures 1c shows the segmentation results obtained using the same color features and smoothing parameters as above, and applying to the network the Huang's initialization and the deterministic evolution described in section 3 (instead of simulated annealing). The results obtained using Algoritm 1 and Algorithm 2 are shown in figures 1d and 1e. The experimental results are summarized in Table I for a quantitative comparison of the performance of the different segmentation algorithms. The performance of the devised algorithms are comparable for all the images analyzed, however the computational cost of Algorithm 2 is lower.

Segmented image	number of regions	average area	Average color error	Evaluation measure (f)
1b	30711	17.65	6.14	181.70
1c	269	243.62	10.38	225.94
1d	414	158.29	8.74	121.75
1e	246	266.40	8.83	109.47

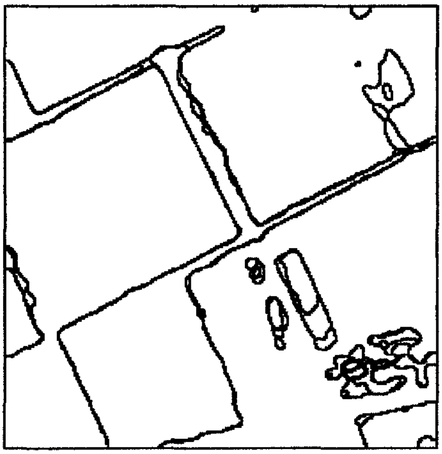
Table I: Summary of experimental results.

6. References

1. J.J. Hopfield "Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79: 2554-2558 (1982).
2. J.J. Hopfield "Neurons with graded response have collective computational properties like those of two-states neurons". Proc. Natl. Acad. Sci. USA 81: 3088-3092 (1984).
3. C-L. Huang "Parallel image segmentation using modified Hopfield model" *Pattern Recognition Letters* 13: 345-353 (1992).
4. A. Rosenfeld, A.C. Kak "Digital picture processing" Academic Press Inc, Orlando, Florida, 1982.
5. Y-H Yang, J. Liu "Multiresolution image segmentation" *IEEE Trans. on Pattern Analysis and Machine Intelligence* 16: 689-700 (1994).
6. Y. Takefuji "Neural network parallel computing", Kluwer Academic Publishers, 1992.
7. A.P. Witkin "Scale-Space Filtering: A New Approach to Multi-scale Description" *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* 3: 39A.1.1-39A.1.4 (1984).

a)

b)



c)

d)

e)

Fig. 1: a) Aircraft photograph digitalized with a flat bed scanner. **b)** Contours of segmentation results obtained by histogram thresholding. **c)** Contours of segmentation results obtained with Algorithm 1 using Huang's initialization. **d)** Contours of segmentation results obtained with Algorithm 1. **e)** Contours of segmentation results obtained with Algorithm 2.