A reduced algorithm from Faugeras-Berthod's theorem in labeling problems

Crescenzio Gallo*

Giancarlo De Stasio†

Dipartimento di Scienze Economiche, Matematiche e Statistiche

Università di Foggia, Italy

Via IV Novembre, 1 - 71100 Foggia

Abstract

This paper illustrates a new approach to labeling ("object classification") problems, and it targets the simplification of a (computationally) complex algorithm based on Faugeras and Berthod's theorem.

Keywords labeling, edge detection, probabilistic algorithms, pixel classification.

*e-mail: *c.gallo@unifg.it*

†e-mail: *g.destasio@unifg.it*

1 Introduction

Our work aims to study the possible applications of Baum-Eagon inequality [3] to the "labeling" problems, which consist in assigning classes (labels) to objects. For example, let us consider an image whose included objects' contours we want to outline (*edge detection*). In this case, the objects are pixels of which the image is made of, and the labels (classes) assignable to every pixel can be "contour pixel", "not-contour pixel".

Many authors have faced this problem; in particular, Faugeras and Berthod [1] require every object to be related with one or more neighbor ones. This situation can be represented by a graph, in which nodes are objects and edges represent existing relations [2] between objects. Such concept can be exemplified considering a phrase containing an ambiguous word: to get its meaning, it may suffice to understand the meaning of neighbor words (*context*). The fact that a word in the phrase allows to go back to the ambiguous word's meaning shows a certain relation between them. Generally, in a phrase the words nearest to the ambiguous one are those useful for its meaning's discovery.

The assignment of a label to an object depends on the labels currently assigned to the related objects: in other words, the context of the object under examination is taken into account. To formalize all this, let us consider, at the beginning, N objects $a_1, a_2, \ldots, a_N$ and L labels $\lambda_1, \lambda_2, \ldots, \lambda_L$. It is necessary to suppose to be able to define a set of initial probabilities, which represent the probability of assigning each label to an object. Elements of such a set are indicated by $p_i(\lambda_k)$, for $i = 1, \ldots, N$ and $k = 1, \ldots, L$, and represent the probability to assign the label λ_k to the object i.

Contextual Faugeras and Berthod's information is represented by a conditional probability set $p_{i,j}(\lambda_k|\lambda_l)$, where $i, j = 1, \ldots, N$ and $k, l = 1, \ldots, L$, representing the probability of assigning label λ_k to the object i, currently having neighbor object j assigned label λ_l. The object j must belong to the set $V_i(\lambda_k)$, which is the set of objects related to i, the object currently having label λ_k assigned to it. In many applications, objects related to a specific one do not depend on the label currently assigned to it; in such a case, the set $V_i(\lambda_k)$ will be simply denoted by V_i (aka "homogeneous case"). In practical problems, the initial probabilities suffer from two lacks, i.e.:

1. **Inconsistency**. In practice, they do not verify the relationship

$$p_i(\lambda_k) = \sum_{j \in V_i(\lambda_k)} \sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l) p_j(\lambda_l) \tag{1}$$

   In other words, initial probabilities are not compatible with conditional probabilities.

2. **Ambiguity**. The initial probabilities are ambiguous if, for at least one $i = 1, \ldots, N$, there exists at least one $l = 1, \ldots, L$ such as vector $\overline{p}_i = [p_i(\lambda_1), \ldots, p_i(\lambda_l), \ldots, p_i(\lambda_L)] \neq [0, \ldots, 1, \ldots, 0]$ (i.e., there is an ambiguity for an object when it tends to fall in more than one class).

2 Consistency and ambiguity functions

Faugeras and Berthod define two functions C_1 and C_2 measuring, respectively, consistency and ambiguity. Consistency is measured through the formula:

$$C_1 = \frac{1}{2N} \sum_{i=1}^{N} \|\overline{p}_i - \overline{q}_i\|^2$$

where $\overline{q}_i$ is a vector having, for each i, the form $[q_i(\lambda_1), q_i(\lambda_2), \ldots, q_i(\lambda_L)]$. Fixed i and k, the values $q_i(\lambda_k)$ are given by the following formula:

$$q_i(\lambda_k) = \frac{Q_i(\lambda_k)}{\sum_{l=1}^{L} Q_i(\lambda_l)}$$

where

$$Q_i(\lambda_k) = \frac{1}{|V_i(\lambda_k)|} \sum_{j \in V_i(\lambda_k)} \sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l) p_j(\lambda_l).$$

The values $q_i(\lambda_k)$ represent an estimate of the probability $p_i(\lambda_k)$ on the basis of the set of conditional probabilities $p_{ij}(\lambda_k|\lambda_l)$ [1]; Faugeras and Berthod's consistency is guaranteed by the equivalence $p_i(\lambda_k) = q_i(\lambda_k)$[1]. From this, the aim is to minimize the function C_1 (which just represents the Euclidean distance between $\overline{p}_i$ and $\overline{q}_i$). The factor $\frac{1}{2N}$ is for bounding C_1 between 0 and 1.

Ambiguity is measured through the following function:

$$C_2 = \frac{L}{L-1} \left[1 - \frac{1}{N} \sum_{i=1}^{N} \|\overline{p}_i\|^2 \right]$$

where $\overline{p}_i$ is the probability vector $[p_i(\lambda_1), p_i(\lambda_2), \ldots, p_i(\lambda_L)]$. Let us observe that in C_2 the factor in square brackets represents the entropy function; the factor $\frac{L}{L-1}$ also here serves to bound C_2 between 0 and 1. Entropy function has its minimum when vector $\overline{p}_i = [p_i(\lambda_1), p_i(\lambda_2), \ldots, p_i(\lambda_L)] =$

[1]Tipically, in image processing problems holds the "homogeneous" case, and then $V_i(\lambda_k) = V_i$

$[0, \ldots, 1, \ldots, 0]$, i.e. it is totally unambiguous. In this case, too, the aim is to find C_2's minimum because it guarantees a non-ambiguous labeling. From C_1 and C_2 derives the function named *Global Criterion* $C = \alpha C_1 + (1-\alpha) C_2$, where $0 \leq \alpha \leq 1$. The value α is a constant which represents the relative weight we want to assign to C_1 and C_2; an higher value of α favours C_1 (i.e. consistency), vice versa C_2 (ambiguity).

The search for C's minimum represents the "weak point" of Faugeras and Berthod's algorithm, because this is implemented with the gradient projection method [5] and requires quite complex operations, as well as a relatively high computational cost. More precisely, the algorithm passes from a labeling x_n to the next x_{n+1} according to the formula:

$$x_{n+1} = x_n + \rho_n u_n$$

where u_n is the negative of C's gradient in x_n, and ρ_n is a positive number calculated in such a way to minimize $C(x_{n+1})$. In this case the problem consists in the fact that the searched minimum is linearly bounded by

$$\begin{cases} \sum_{k=1}^{L} p_i(\lambda_k) = 1 \text{ (i=1,\ldots,N)} \\ p_i(\lambda_k) \geq 0 \end{cases} \tag{2}$$

This involves the computation, at every iteration, of a projection operator P_n. The computation of P_n becames necessary because the negative of u_n gradient may point out of (2) hyperplane.

The complexity of Faugeras and Berthod's algorithm leads to a difficult implementation, even with the use of parallel computing architectures; in fact, the work done by every single processor remains heavy. Its complexity becames high especially in the non-homogeneous case (though this last is rarely applied) in which same authors do not define the number of computations necessary to obtain ρ_n. So, we aim to simplify the algorithm's complexity, exploiting Baum-Eagon's theorem. It applies to homogeneous polynomials of degree d; another theorem (Baum-Sell [4]) removes this limitation.

Baum-Eagon's Theorem [4]: Let $P(x) = P(\{x_{ij}\})$ an homogeneous polynomial with nonnegative coefficients in variables $\{x_{ij}\}$ verifying

$$x_{ij} \geq 0, \qquad \sum_{j=1}^{L} x_{ij} = 1, \qquad i = 1, \ldots, N.$$

Let

$$F(x_{ij}) = \frac{x_{ij} \frac{\partial P}{\partial x_{ij}}(x)}{\sum\limits_{j=1}^{L} x_{ij} \frac{\partial P}{\partial x_{ij}}(x)}$$

Then $P(F(x)) > P(x)$ until $F(x) = x$.

Faugeras and Berthod's C function is a quasi-homogeneous polynomial[2] of degree two, if we consider the *homogeneous* case (see section 1). In practice, because C polynomial does not generally have nonnegative coefficients (as required by the previous theorem), it is necessary to transform C in such a manner that the theorem be applied to another polynomial C' with nonnegative coefficients.

Formally, polynomial C has the form:

$$C = \sum_{i,j} \sum_{k,l} k_{ijkl} x_{ik} x_{jl}$$

[2]Except for a constant, which disappears after the application of the partial derivatives $\frac{\partial P}{\partial x_{ij}}(x)$

where k_{ijkl} are the (not all nonnegative) coefficients of the polynomial, x_{ik} $(i, j = 1, \ldots, N; k, l = 1, \ldots, L)$ are the unknown factors of the polynomial. Baum-Eagon's theorem leads to an increasing transformation, so as it searches relative maximum points. The case of C is different, because we must minimize instead of maximize. So, instead of minimizing C, we equivalently maximize $-C$. Then, let:

$$C^{(-)} = -C = -\sum_{i,j} \sum_{k,l} k_{ijkl} x_{ik} x_{jl}.$$

$C^{(-)}$ is still a polynomial with not all nonnegative coefficients. It is possible to make $C^{(-)}$'s coefficients nonnegative increasing each coefficient k_{ijkl} by the quantity

$$m = \min \left\{ \min_{i,j,k,l} \{k_{ijkl}\}, 0 \right\} \tag{3}$$

so $C^{(-)}$ becames:

$$\begin{aligned} C^{(T)} &= -\sum_{i,j} \sum_{k,l} (k_{ijkl} + m) x_{ik} x_{jl} \\ &= -\sum_{i,j} \sum_{k,l} (k_{ijkl} x_{ik} x_{jl} + m x_{ik} x_{jl}) \\ &= -\sum_{i,j} \sum_{k,l} k_{ijkl} x_{ik} x_{jl} - m \sum_{i,j} \sum_{k,l} x_{ik} x_{jl} \\ &= C^{(-)} - mN^2 \end{aligned}$$

where it is to be considered that holds the relation: $\sum_{k=1}^{L} x_{ik} = 1$, for each $i = 1, \ldots, N$.

Applying Baum-Eagon's theorem to $C^{(T)}$, we have:

$$C^{(T)}(x) \leq C^{(T)}(F(x))$$

from which:

$$\left[C^{(-)}(x) - mN^2\right] \leq \left[C^{(-)}(F(x)) - mN^2\right]$$
$$C^{(-)}(x) \leq C^{(-)}(F(x))$$
$$C(x) \geq C(F(x))$$

The above steps then show that, shifting polynomial coefficients by a constant quantity and applying the theorem, we obtain the growing of both $C^{(T)}$ and $C^{(-)}$, which coincides with the decreasing of the Faugeras and Berthod's C function.

3 Experimental results

Baum-Eagon's theorem application requires, from a practical point of view, the writing of the function C in polynomial form. The development of the function

$$C = \alpha C_1 + (1-\alpha) C_2$$

(see appendix A) leads to the following (quasi-homogeneous) polynomial form:

$$\begin{aligned}
C = {} & \frac{3\alpha L - \alpha - 2L}{2N(L-1)} \sum_{i=1}^{N} \sum_{k=1}^{L} x_{ik}^2 \\
& - \frac{\alpha}{N} \sum_{i=1}^{N} \sum_{k=1}^{L} x_{ik} \sum_{j \in V_i} \sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l) x_{jl} \\
& + \frac{\alpha}{2N} \sum_{i=1}^{N} \sum_{k=1}^{L} x_{ik} \sum_{j \in V_i} \sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l) x_{jl} \sum_{\omega \in V_i} \sum_{v=1}^{L} p_{\omega i}(\lambda_v|\lambda_k) \\
& + M
\end{aligned}$$

where:

- L is the number of labels,
- N is the number of objects,
- V_i is the set of objects related to the object i[3],
- $M = \frac{(1-\alpha)L}{L-1}$,
- $x_{ik} = p_i(\lambda_k)$.

From the above we note the quasi-homogeneity of C (in the variables x_{ik}) apart of constant M, that does not influence the computation of the partial derivatives, used for the implementation of the algorithm.

The algorithm which implements the above mentioned method must initially find a constant value such as it makes all polynomial coefficients non-negative, so obtaining a new polynomial to which is possible to apply Baum-Eagon theorem. Such a constant value is obtained by increasing every coefficient by a quantity equal to the minimum m as in (3).

Labeling through Baum-Eagon theorem has been tested on the *thresholding* problem. This one consists in the transformation of an image, made of a matrix of different grey level tones pixels, into another image made only of black and white pixels. Formalizing the problem, the image is made of $n \times m$ objects and $L = 2$ labels, corresponding to "*light pixel*" and "*dark pixel*". The initial probability set is computed according to a method suggested by Rosenfeld and Russel [7]: let d and l be, respectively, the toward "*dark*" and toward "*light*" grey levels; let z_i be the grey level of the i-th pixel. Then, for

[3]We assume the "homogeneous" case, so $V_i(\lambda_k) = V_i$

that pixel we'll have the following initial probabilities: $p_{i,dark} = (l - z_i)/(l - d)$ and $p_{i,light} = (z_i - d)/(l - d)$.

As far as the conditional probabilities set is concerned, Rosenfeld and Peleg [8] suggest a method based on statistical computation. Initially, there is an estimate of the probability that every pixel has a certain label λ; this is realized through the formula:

$$\overline{P}(\lambda) = \frac{1}{N} \sum_{(x,y)} P_{(x,y)}(\lambda)$$

where N is the number of pixels, and pairs (x, y) are the coordinates of every pixel. Then, it is computed the joint probability of every pair (x, y) and $(x + i, y + j)$ of neighbor points have assigned, respectively, labels (λ, λ') according to the formula:

$$P_{ij}(\lambda, \lambda') = \frac{1}{N} \sum_{(x,y)} P_{(x,y)}(\lambda) P_{(x+i,y+j)}(\lambda')$$

From the two above derives the conditional probabilities formula:

$$P_{ij}(\lambda|\lambda') = \frac{P_{ij}(\lambda, \lambda')}{\overline{P}(\lambda')}$$

which has been used in our experimental tests to set the initial probabilities.

4 Conclusion

The goodness of Baum-Eagon approach has been hypotetized in various environments, especially in probabilistic labeling problems, well suited to AI with parallel computing architecture. Starting with Faugeras and Berthods overly computationally complex algorithm, we developed a simplified version using Baum-Eagon inequality, and reached positive experimental results, which encourage us to refine and carry on trying more complex test sets.

In Appendix A is reported the detailed development of the Baum-Eagon C function in a quasi-homogeneous polynomial form of degree two, and in Appendix C the experimental results obtained are illustrated.

A Development of the Baum-Eagon C function

In order to obtain the final form of $C = \alpha C_1 + (1-\alpha)C_2$ we first consider

$$C_1 = \frac{1}{2N}\sum_{i=1}^{N}\|p_i - q_i\|^2$$

Developing C_1 we obtain:

$$\begin{aligned}
C_1 &= \frac{1}{2N}\sum_{i=1}^{N}\|p_i - q_i\|^2 \\
&= \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}\left[p_i(\lambda_k) - q_i(\lambda_k)\right]^2 \\
&= \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}\left[p_i(\lambda_k) - \frac{\frac{1}{|V_i|}\sum_{j\in V_i}\sum_{l=1}^{L}p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)}{\sum_{m=1}^{L}\frac{1}{|V_i|}\sum_{j\in V_i}\sum_{l=1}^{L}p_{ij}(\lambda_m|\lambda_l)p_j(\lambda_l)}\right]^2 \\
&= \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}\left[p_i(\lambda_k) - \sum_{j\in V_i}\sum_{l=1}^{L}p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)\right]^2
\end{aligned}$$

In the last step we made use of the relation: $\sum_{m=1}^{L}\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_m|\lambda_l)p_j(\lambda_l) = 1$. Now, developing the expression in the above square brackets:

$$C_1 = \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}[p_i(\lambda_k)^2 - 2p_i(\lambda_k)\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)$$
$$+\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)\sum_{\omega\in V_i}\sum_{v=1}^{L} p_{i\omega}(\lambda_k|\lambda_v)p_\omega(\lambda_v)]$$

and then:

$$C_1 = \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)^2 - \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)\sum_{j\in V_i}\sum_{l=1}^{L} pij(\lambda_k|\lambda_l)p_j(\lambda_l)$$
$$+\frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)\sum_{\omega\in V_i}\sum_{v=1}^{L} p_{i\omega}(\lambda_k|\lambda_v)p_\omega(\lambda_v)$$

In order to obtain a form of C_1 more suited to the partial derivatives needed for the development of our algorithm, let us consider the case of image processing (which we are treating in our work). In this situation, from the Euclidean distance it is obvious to derive a sort of “reciprocity” in the sequence of indexes involved in the formula, in the sense that if a pixel i is at distance d from a pixel j, obviously the same holds for j respect to i.

So we may write the final form of C_1:

$$C_1 = \frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)^2 - \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)\sum_{j\in V_i}\sum_{l=1}^{L} pij(\lambda_k|\lambda_l)p_j(\lambda_l)$$
$$+\frac{1}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)\sum_{\omega\in V_i}\sum_{v=1}^{L} p_{\omega i}(\lambda_v|\lambda_k)p_i(\lambda_k)$$

Let us consider the development of C_2. We obtain:

$$C_2 = \frac{L}{L-1}\left[1 - \frac{1}{N}\sum_{i=1}^{N}\|\overline{p}_i\|^2\right]$$
$$= \frac{L}{L-1}\left[1 - \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)^2\right]$$

Multiplying C_1 and C_2 respectively by α and $(1-\alpha)$ we obtain the polynomial form of “Global Criterion” C function. Developing in detail we have:

$$\begin{aligned}
C &= \alpha C_1 + (1-\alpha)C_2 \\
&= \frac{\alpha}{2N}\sum_{i=1}^{N}\|\overline{p}_i - \overline{q}_i\|^2 + (1-\alpha)\frac{L}{L-1}\left[1 - \frac{1}{N}\sum_{i=1}^{N}\|\overline{p}_i\|^2\right] \\
&= \frac{\alpha}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)^2 - \frac{\alpha}{N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l) \\
&\quad + \frac{\alpha}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)\sum_{\omega\in V_i}\sum_{v=1}^{L} p_{\omega i}(\lambda_v|\lambda_k)p_i(\lambda_k) \\
&\quad + \frac{(1-\alpha)L}{L-1} - \frac{(1-\alpha)L}{N(L-1)}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)^2 \\
&= \frac{3\alpha L - \alpha - 2L}{2N(L-1)}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)^2 \\
&\quad - \frac{\alpha}{N}\sum_{i=1}^{N}\sum_{k=1}^{L} p_i(\lambda_k)\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l) \\
&\quad + \frac{\alpha}{2N}\sum_{i=1}^{N}\sum_{k=1}^{L}\sum_{j\in V_i}\sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l)p_j(\lambda_l)\sum_{\omega\in V_i}\sum_{v=1}^{L} p_{\omega i}(\lambda_v|\lambda_k)p_i(\lambda_k) \\
&\quad + \frac{(1-\alpha)L}{L-1}
\end{aligned}$$

From the above formula it is simple to get the partial derivatives:

$$\frac{\partial C}{\partial p_i(\lambda_k)} = \frac{3\alpha L - \alpha - 2L}{N(L-1)} p_i(\lambda_k) - \frac{\alpha}{N} \sum_{j \in V_i} \sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l) p_j(\lambda_l) + \frac{\alpha}{2N} \sum_{j \in V_i} \sum_{l=1}^{L} p_{ij}(\lambda_k|\lambda_l) p_j(\lambda_l) \sum_{\omega \in V_i} \sum_{v=1}^{L} p_{\omega i}(\lambda_v|\lambda_k)$$

B Glossary

Ambiguous (labeling): A labeling is ambiguous when, for each object, there is no certainty to assign a label λ to it. Formally, exists at least one object i and a label λ for which results $\mathrm{p}_i(\lambda) \neq 1$.

Entropy: a function used in thermodynamics and in information theory. In information theory it measures the average amount of information contained in a statistical set of messages.

Inconsistent (labeling) [1]: a labeling is inconsistent if, on the basis of conditional probabilities, is obtained an estimate $q_i(\lambda_k)$ of the probabilities $p_i(\lambda_k)$ (for $i = 1, \ldots, N$ number of objects, and $k = 1, \ldots, L$ number of labels) so that it results $q_i(\lambda_k) \neq p_i(\lambda_k)$. In other words, the set of conditional probabilities is not coherent with the set of probabilities $p_i(\lambda_k)$.

Object: any entity classifyable in one of L distinct classes.

Homogeneous (case): is the case in which, given an object i, the set of objects related to i does not depend on the label currently assigned to i.

Label: one of the possibile classes that may be assigned to an object.

Labeling: process consisting in the assignment of L classes (labels) to N objects.

Pixel: the most basic component of any digital image.

Homogeneous polynomial of degree d: is a polynomial whose monomials are all of degree d.

Initial probabilities: is a set of initially assignable probabilities, that suffer from ambiguity and inconsistency problems. In [8] are illustrated some statistical methods of initial probabilities computation.

Relation (between objects): an object i is related to another object j if the probabilities $p_i(\lambda_k)$ ($k = 1, \ldots, L$ number of labels) depend on the label currently assigned to the object j.

C Experimental results

We got our experimental results using a dual-processor 2GHz PowerMac G5 with 1GB of RAM and Mac OS X 10.3 Operating System; the software has been implemented in Gnu C. We aim to completely rewrite our code so as to parallelize and test it on significant data sets (derived from bitmap images), to strength and verify our "simplified" algorithm.

Here follows a list of tests operated on sample images, with a few significant iterations showing the ongoing labeling process (”dark” or ”light” progressive assignment to image pixels, depending on neighbor labels).

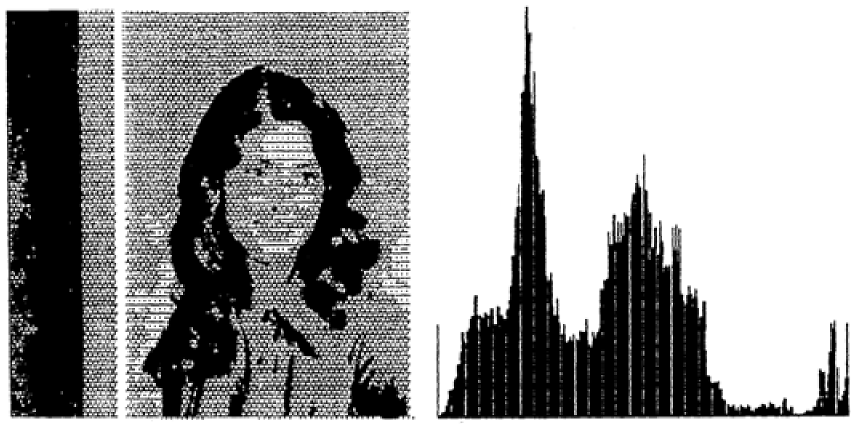

Figure 1: Test 1 - Initial labeling and iteration #1

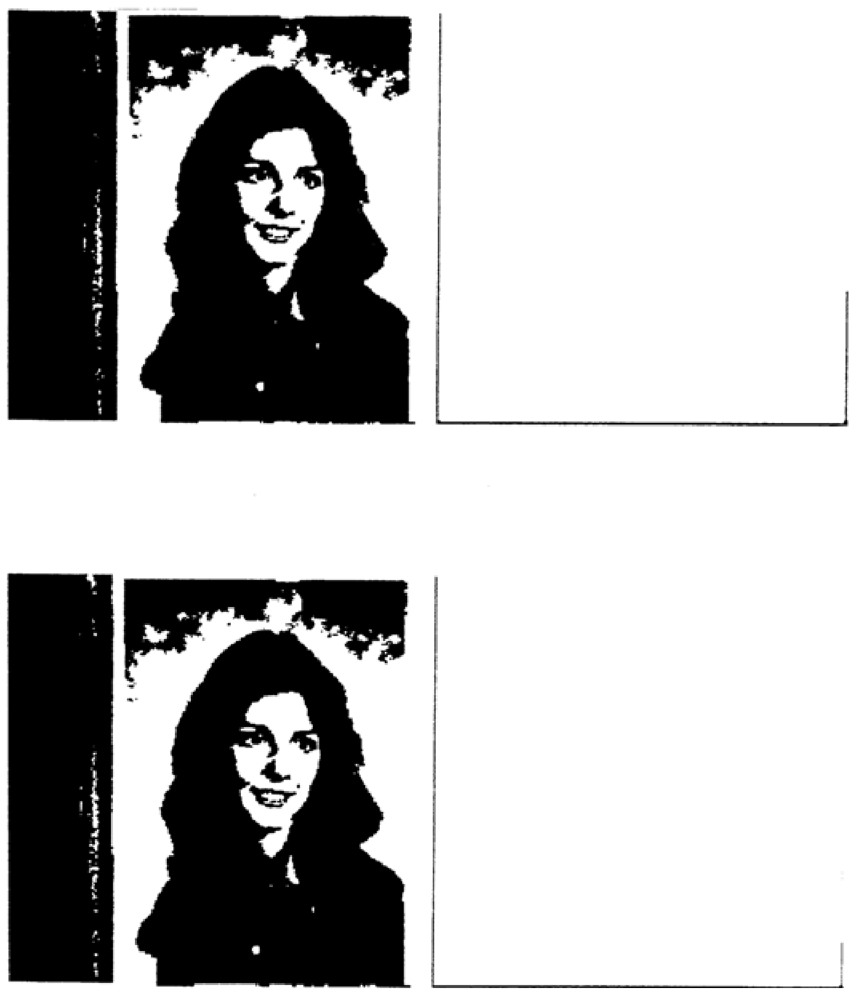

Figure 2: Test 1 - Iterations #16 and 17

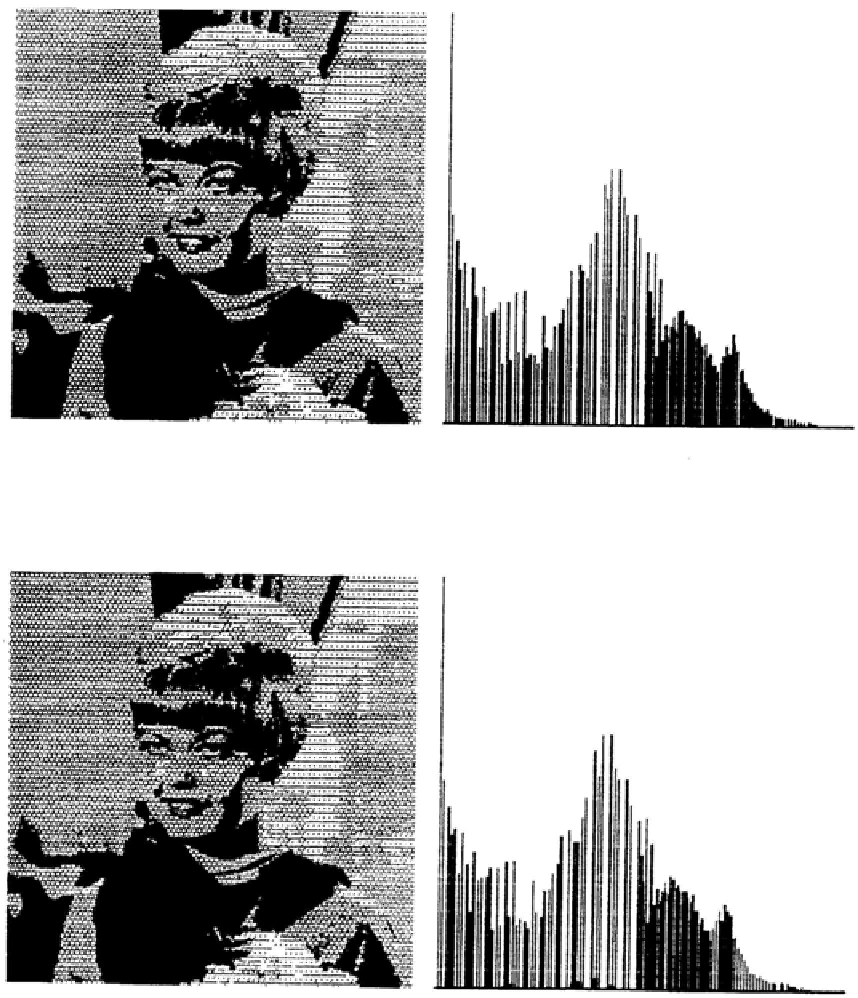

Figure 3: Test 2 - Initial labeling and iterations #1

Figure 4: Test 2 - Iterations #4, 5 and 10

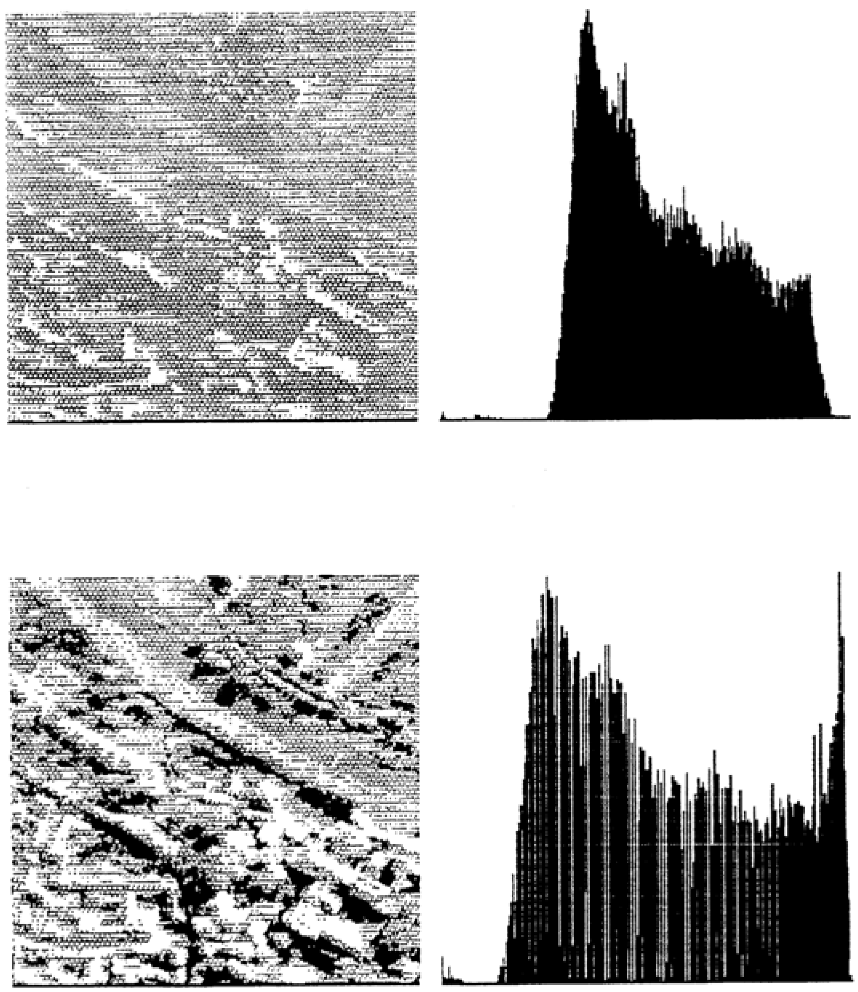

Figure 5: Test 3 - Initial labeling and iteration #1

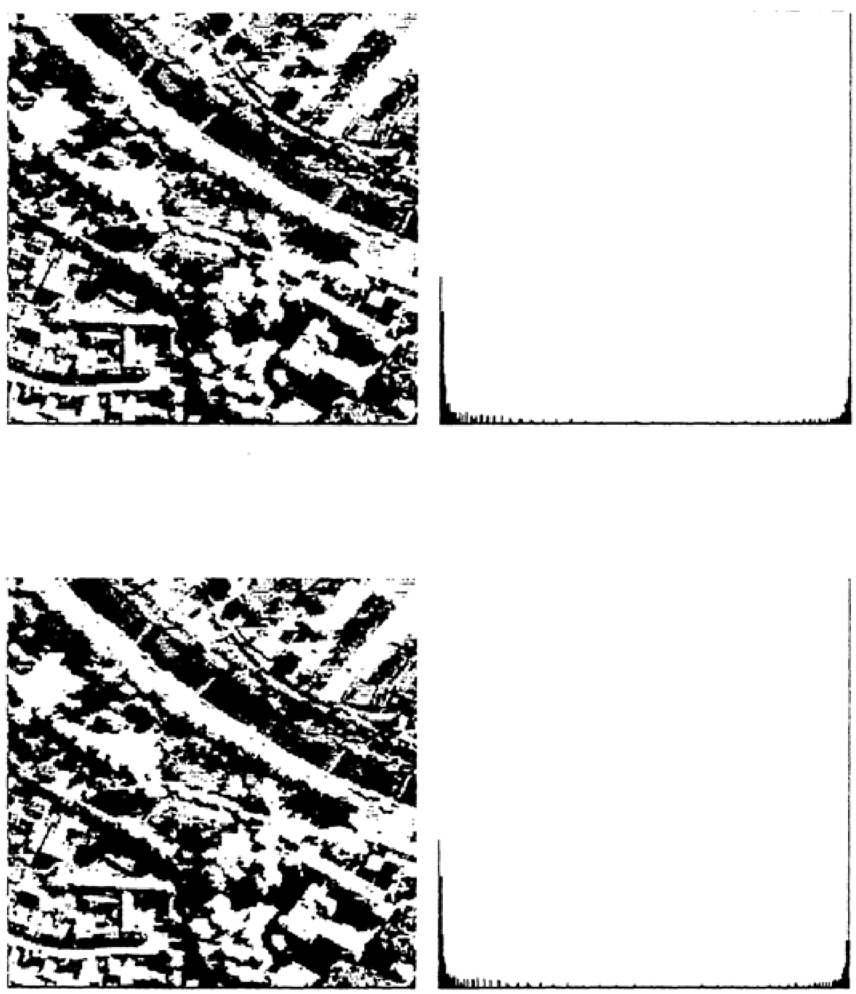

Figure 6: Test 3 - Iterations #4 and 5

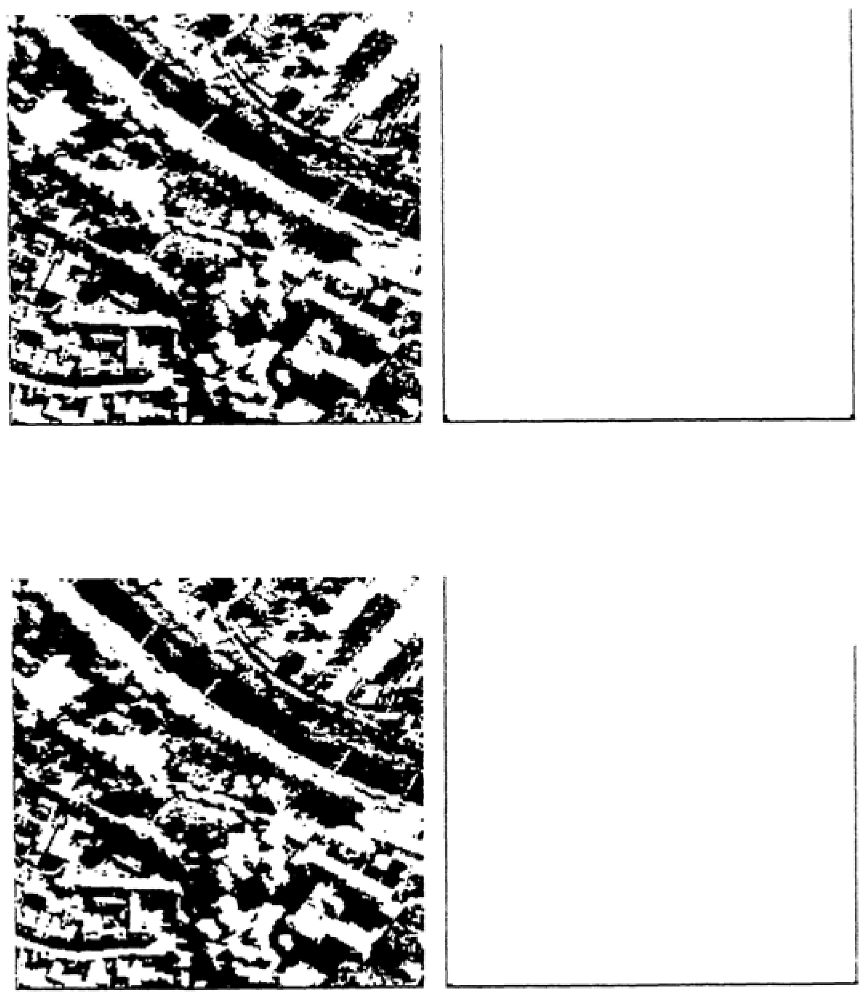

Figure 7: Test 3 - Iterations #8 and 9

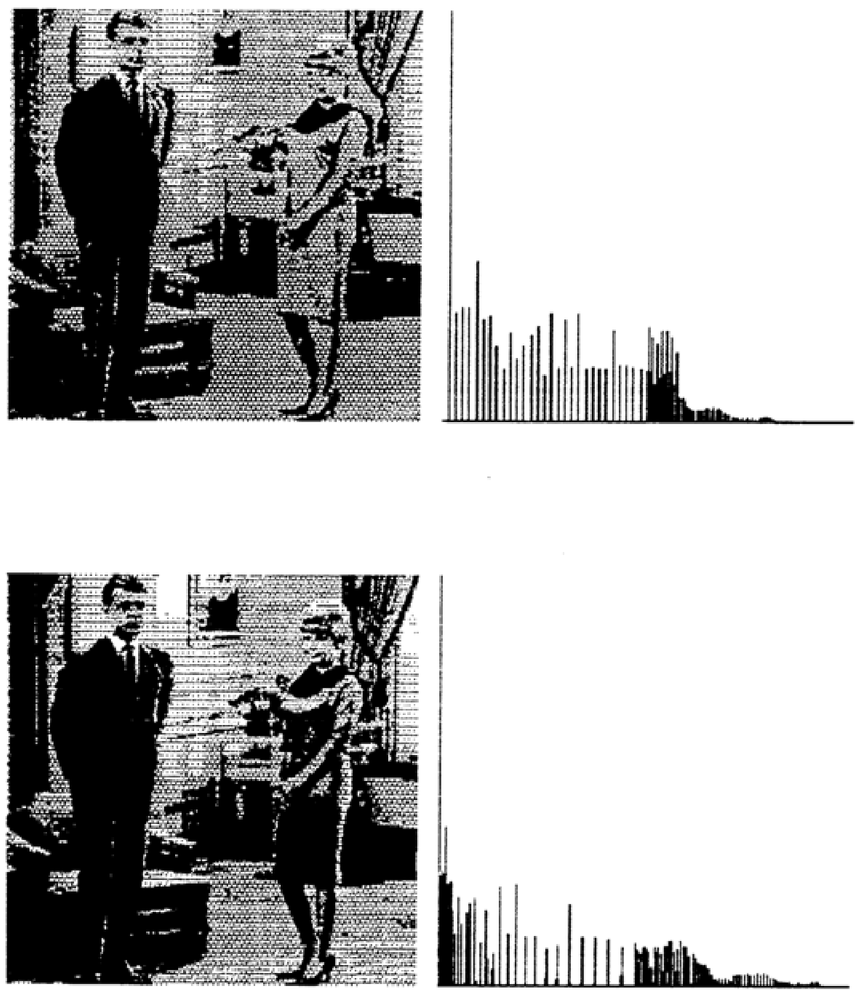

Figure 8: Test 4 - Initial labeling and iteration #2

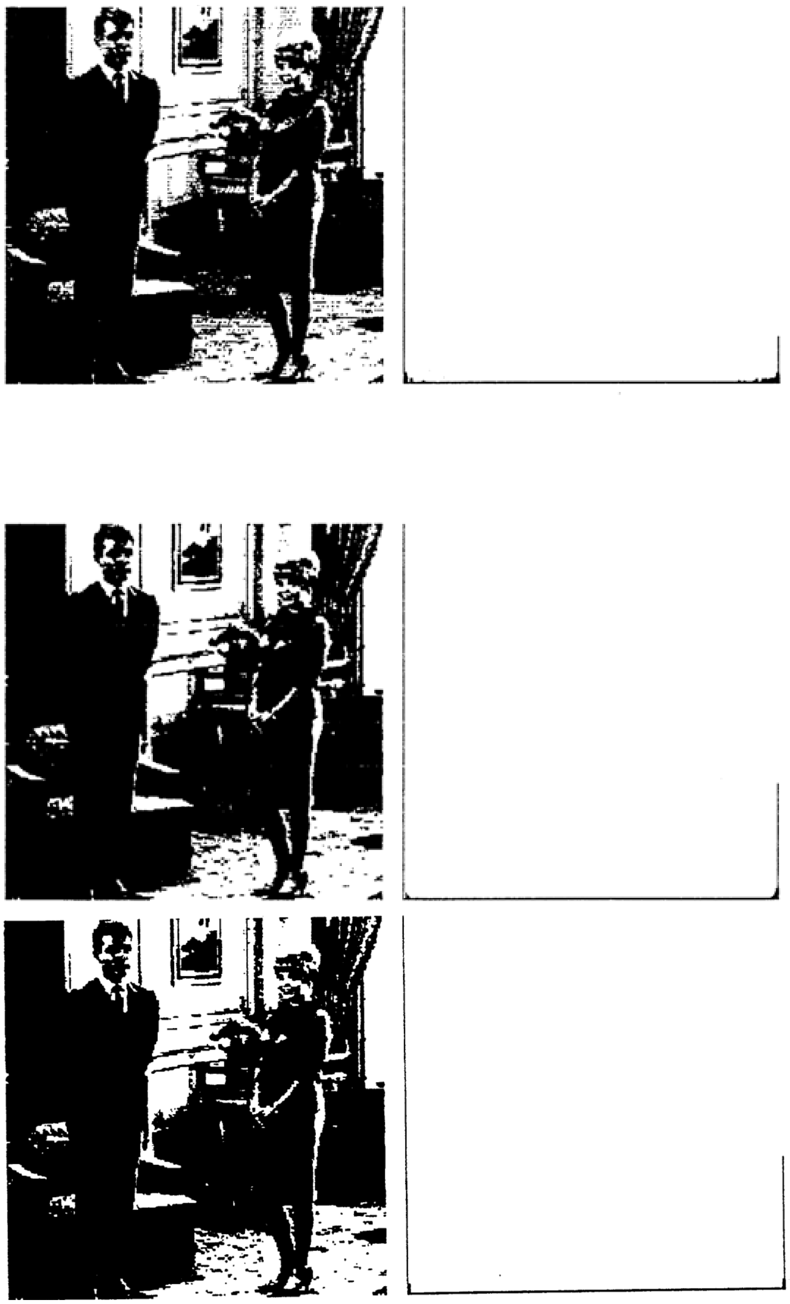

Figure 9: Test 4 - Iterations #10, 14 and 16

References

[1] O. D. Faugeras and M. Berthod, *Improving consistency and reducing ambiguity in stochastic labeling: an optimization approach*, IEEE Trans. Patt. Anal. and Mach. Intell., Vol. PAMI-3, No. 4, pp. 412–424, July 1981.

[2] R. A. Hummel and S. W. Zucker, *On the foundations of relaxation labeling processes*, IEEE Trans. Patt. Anal. and Mach. Intell., Vol. PAMI-5, No. 3, pp. 267–286, May 1983.

[3] L. E. Baum and J. A. Eagon, *An inequality with application to statistical estimation for probabilistic function of Markov processes and to a model for ecology*, Bull. Amer. Math. Soc., Vol. 73, pp. 361–363, 1967.

[4] L. E. Baum and G. R. Sell, *Growth transformation for functions on mainfolds*, *Pacific Journal of Mathematics*, Vol. 27, No. 2, pp. 211–227, 1968.

[5] J. B. Rosen, *The gradient projection method for nonlinear programming-Part I: Linear Constraints*, *J. Soc. Appl. Math.*, Vol. 8, pp. 181–217, Mar. 1960.

[6] A. Rosenfeld and A. C. Kak, *Digital picture processing*, *Academic Press New York*, pp. 55–187, 1976.

[7] A. Rosenfeld and R. C. Smith, *Thresholding using relaxation*, *IEEE Trans. Patt. Anal. and Mach. Intell.*, Vol. PAMI-3 No. 5, pp. 598–606, Sept. 1981.

[8] S. Peleg and A. Rosenfeld, *Determining compatibility coefficients for curve enhancement relaxation processes*, *IEEE Trans. Syst., Man. and Cybernetics*, Vol. SMC-8, No. 7, pp. 548–555, July 1978.

[9] P. K. Sahoo, S. Soltani and A. K. C. Wong, *A survey of thresholding techniques*, *Computer Vision, Graphics, and Image Processing*, Vol. 41, pp. 233–260, 1988.

[10] M. Pelillo, *Nonlinear relaxation labeling: a growth transformation*, *Proc. 12th Int. Conf. Pattern Recognition*, Jerusalem, Israel, Oct. 1994.

[11] M. Pelillo and M. Refice, *Learning compatibility coefficients for relaxation labeling processes*, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 16, No. 9, pp. 933–945, Sept. 1994.

[12] R. M. Haralick and L. G. Shapiro, *The consistent labeling problem: Part I*, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-1, No. 2, pp. 173–184, 1979.

[13] C. Gallo and G. Mainardi, *Integrazione di classificatori in criteri di riconoscimento stocastico*, Quaderni di Statistica, Facoltà di Economia, Università di Bari, 1996.

[14] Zen Chen, Shaw-Yin Lin and Yi-Yao Chen, *A parallel architecture for probabilistic relaxation operations on images*, Pattern Recognition, vol.23 n.6, pp.637–643, 1990.

[15] R. M. Haralick, J. L. Mohammed and S. W. Zucker, *Compatibilities and the fixed points of arithmetic relaxation processes*, Comput. Graph. Image Processing, vol.13, pp.242–256, 1980.

[16] D. G. Luenberger, *Linear and Nonlinear Programming*, Reading, MA, Addison-Wesley, 1984.

[17] J. Illington and J. Kittler, *Optimisation algorithms in probabilistic relaxation labelling*, Proc. of the NATO Advanced Study Institute on Pattern recognition theory and applications, p.109-117, Spa-Balmoral, Belgium, January 1987.

[18] J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1991.

[19] J. Kittler and J. Illingworth, *Relaxation labelling algorithms-a review*, Image and Vision Computing, vol.3 n.4, pp.206–216, Nov. 1986.

[20] X. Zhuang, R. M. Haralick and H. Joo, *A Simplex-Like Algorithm for the Relaxation Labeling Process*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.11 n.12, pp.1316–1321, December 1989.

[21] J. Lin, *Divergence measures based on the Shannon entropy*, IEEE Trans. Inform. Theory, vol. IT-37, no. 1, pp. 145–151, 1991.

[22] M. Pelillo and M. Refice, *An optimization algorithm for determining the compatibility coefficients of relaxation labeling processes*, Proc. 11th Int. Conf. Pattern Recognit., The Hague, The Netherlands, pp. 145–148, 1992.

[23] M. Pelillo and M. Refice, *Learning compatibility coefficients for word-class disambiguation relaxation processes*, Proc. Int. Conf. Spoken Language Processing, Banff, Canada, pp. 389–392, 1992.

[24] J. L. Mohammed, R. A. Hummel and S. W. Zucker, *A gradient projection algorithm for relaxation methods*, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-5, no. 3, pp. 330–332, 1983.

[25] P. Parent and S. W. Zucker, *Radial Projection: An Efficient Update Rule for Relaxation Labeling*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.11 n.8, pp.886–889, August 1989.

[26] J. J. Hopfield and D. W. Tank, *Neural computation of decisions in optimization problems*, Biol. Cybern., vol. 52, pp. 141–152, 1985.

[27] T. A. Jamison and R. J. Schalkoff, *Image labelling: a neural network approach*, Image and Vision Computing, vol.6 n.4, pp.203–214, November 1988.

[28] P. J. M. Laarhoven and E. H. L. Aarts, *Simulated annealing: theory and applications*, Kluwer Academic Publishers, Norwell, MA, 1987.

[29] M. Pelillo and M. Refice, *Syntactic category disambiguation through relaxation processes*, Proc. EUROSPEECH 91, Genova, Italy, pp. 757–760, 1991.

[30] C. T. Gens, *Relaxation and neural learning: points of convergence and divergence*, Journal of Parallel and Distributed Computing, vol.6 n.2, pp.217–244, April 1989.

[31] Shiaw-Shian Yu and Wen-Hsiang Tsai, *Relaxation by the Hopfield neural network*, Pattern Recognition, vol.25 n.2, pp.197–209, Feb. 1992.

[32] T. Elfving and J. O. Eklundh, *Some properties of stochastic labeling procedures*, Comput. Graph. Image Processing, vol. 20, pp. 158–170, 1982.

Biography

Crescenzio Gallo was born in Carapelle (FG), Italy, in 1956. He received the Computer Science bachelor's degree (with honors) from the University of Bari, Italy, in 1978. During 1978-1980, he stayed with the Istituto di Scienze dell'Informazione (ISI), Bari, Italy as a Research Assistant in Information Systems, and with Telespazio, Rome, Italy participating to Landsat satellite projects of Italian (ASI) and European Space Agency (ESA). From 1982 to 2003 he has been a high school full-time teacher in Computer Science at Foggia, Italy, and since 1993 he has been a contract professor of Computer Science at the University of Foggia, Italy. Since January 2004 he is an Assistant Professor at the Dept. of Economic, Mathematical and Statistical Sciences, University of Foggia, Italy. His primary research interests include information theory and its economic applications, with special emphasis on (wireless) networks. Dr. Gallo is an IEEE Member since 1998, and an ACM professional member, UMI and AMASES associate since 2004.

Giancarlo De Stasio was born in Foggia, Italy, in 1967. He received the Computer Science bachelor's degree from the University of Bari, Italy, in 1994. Since 1998 he has been applyed with the University of Foggia, Italy, as a consultant in Computer Science and statistical projects. His research interests are now focused on discrete algorithms, image processing and databases.