BAYESIAN ADAPTIVE OIL SPILL SEGMENTATION
OF SAR IMAGES VIA GRAPH CUTS

Sónia Pelizzari, José M. Bioucas Dias

Instituto de Telecomunicações, I.S.T., Torre Norte, Piso 10, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

ABSTRACT

This paper presents a Bayesian supervised segmentation algorithm aimed at oil spill detection in SAR images, a crucial step in any SAR based automatic oil spill surveillance system. The data term, i.e., the density of the observed backscattered signal given the region, is modeled as a finite mixture of Gamma distributions. The mixture renders robustness to backscattering fluctuations inside each region. The prior is an M-level Markov Random Field defined on a 2D grid, enforcing local continuity in a statistical sense. The maximum a posteriori (MAP) segmentation is computed efficiently by means of recent graph-cut techniques. The effectiveness of the proposed method is illustrated with real ERS and ENVISAT data.

INTRODUCTION

Many approaches to oil spill detection using SAR images have been proposed in recent years [1, 2, 3]. In the processing chain of such detection systems, segmentation of dark spots is usually one of the first steps to be performed [4]. The algorithms exploit the well known effect of capillary-waves dampening caused by oil films [5]. A common segmentation method in use is adaptive thresholding [1]. This work presents an algorithm for supervised segmentation of dark spots in SAR/ASAR images, following a Bayesian approach

The data model used in this work is a finite Gamma mixture for each class (water and oil) with a given predefined number of components. In fact, this density is well suited to filtered intensity SAR/ASAR images as shown e.g. in [6]. By using a mixture, we aim at describing the continuous backscattering variability that may be observed in the SAR sea data.

To estimate the parameters of the class conditional densities, we develop a new expectation maximization (EM) algorithm. Similar algorithms can be found in the literature [3], but they only infer one parameter for each Gamma class: the mean value \( \mu \). As they fix the number of looks to \( N \), the variance of the distribution is then automatically given by \( \mu^2/N \). This is an approximation that relays on the assumption of conditionally independence of neighboring pixels measurements. We estimate both the mean and the variance for each Gamma distribution in the mixture, rendering the algorithm more adaptable to real measurements.

The prior used to impose local homogeneity is a second order Markov Random Field (MRF), more specifically an Ising Model. To estimate the labels, we maximize its posterior distribution (MAP), and are led to an energy minimizing problem, which we solve efficiently with graph-cut techniques [7].

The article is organized as follows: part 1 gives an overview of the Bayesian approach, describing the data model and the prior and introducing the notation used; part 2 describes the EM algorithm developed for inferring the parameters of the Gamma mixture; part 3 gives an overview of the graph-cut methodology and explains how our problem was mirrored in a graph cut problem; part 4 shows results of applying the segmentation to some images. The tool has been applied to ERS SAR and ENVISAT ASAR images representing well documented oil disasters in the Portuguese and Spanish coast (Douro disaster in 1994 and Prestige disaster in 2002), as well as to simulated images, for error rate assessment.

Finally, part 5 presents concluding and future work remarks.

---

1 The work was supported in part by the Portuguese “Fundação para a Ciência e Tecnologia” (FCT) under the grant PDCTE/CPS/49967/2003 and by the European Space Agency (ESA) under the grant ESA/C1:2422.
1. PROPOSED APPROACH

Let $S := \{1, 2, \ldots, N\}$ be the set of pixels (sites) where measurements are available. Let $\Omega := \{1, 2, \ldots, c\}$ be the set of classes and $x := \{x_i \in \Omega, i \in S\}$ be the segmentation, or labeling, of $S$; i.e. a partition of $S$ into a disjoint set of regions $R_k = \{i \in S, x_i = k\}$ so that $\cup R_k = S$. In this work, we only use two classes, $c = 2$, one for water and one for oil, but the methodology is easily expandable to more classes.

In the following, $y := \{y_1, y_2, \ldots, y_N\}$ is the observed image at sites $S$. By applying a segmentation algorithm to the image $y$, we get $x := \{x_1, x_2, \ldots, x_N\}$, where $\hat{x}_i$ is the corresponding estimated label of pixel $i$. Most of the notation used in the article is adopted from [8].

We assume conditional independence of the measurements $y_i$ and that each $y_i$ has the same known conditional density function $f(y_i|x_i)$, called data model or sensor function. The described record independence is a sensing mechanism property and means that we may express the conditional density of the observed records (1) $y$ as a multiplication of terms, each one depending only on one pixel record, i.e.,

$$I(y | x) = \prod_{i=1}^{n} f(y_i | x_i) \quad (1)$$

The adopted data model is a finite Gamma mixture given by

$$f(y_i | x_i) = \sum_{s=1}^{K} \alpha_s \times f(y_i | x_i, \theta_s) \quad (2)$$

where $K$ is the number of modes in the mixture, $\theta_s$ is the vector of parameters of mode $s$ and $\alpha := (\alpha_1, \ldots, \alpha_K)$ is the vector with the a priori modes probabilities. The density of mode $s$ is given by

$$f(y_i | x_i, \theta_s) = \frac{\lambda_s^{as}}{\Gamma(as)} \times y_i^{as-1} \times \exp(-\lambda_s \times y_i) \quad (3)$$

where parameters $\theta_s := (a_s, \lambda_s)$ and $a_s / \lambda_s^2$ stands for the mean value of (3) and $a_s / \lambda_s^2$ stands for its variance.

A second assumption, that of local smoothness, is modeled by a second order MRF, $P(x)$. The Markov property assumes that

$$P(x_i | x_j, j \in S) = P(x_i | x_j, j \in N_i) \quad (4)$$

where $N_i$ is the set of neighbors of site $i$. Assuming that $P(x_i|x_{N_i}) > 0$, then the Hammersley-Clifford Theorem states that $P(x)$ has the form of the Gibbs distribution,

$$P(x) = \frac{1}{Z} \times \exp(-U(x)) \quad (5)$$

where $Z$ is the so called partition function and $U$ is the energy function

$$U(x) = \sum_{cl \in C} V_{cl}(x) \quad (6)$$

where $C$ is the set of cliques and $V_{cl}(x)$ is the clique potential defined over clique $cl$. Figure 1 illustrates the four 2-order cliques we used.
The Ising Model corresponds to

\[ V(x_r, x_s) = \beta \times \delta(x_r, x_s) \tag{7} \]

where parameter \( \beta \) controls the degree of homogeneity we wish to impose on the scene and \( \delta(x_i, x_j) \) is the discrete delta function. The posterior of the labeling given the observed data is

\[ p(x \mid y) \propto l(y \mid x)P(x) \tag{8} \]

We adopt the MAP criterion to infer \( x \), i.e.

\[ \hat{x} = \arg \max_x p(x \mid y) \tag{9} \]

Fig. 1: 8 pixel neighbourhood (light green) of pixel \( i \) (dark green) with four 2-pixel-cliques represented as blue ellipses.

2. EM ALGORITHM FOR DATA MODEL

When determining the data model to use in equation (8), we need to estimate \( 2 \times K \) Gamma parameters and \( K \) a priori probabilities for each class. We infer \( \theta \) by computing its maximum likelihood (ML) estimate from a training set. The ML estimate is computed via the EM algorithm [9].

The key point in the technique is the introduction of the so called missing data \( z \), such that \( p(y \mid \theta) = \int p(y, z \mid \theta) dz \) and \( p(y, z \mid \theta) \) is easier to manipulate than \( p(y \mid \theta) \). In the particular case of a mixture of densities, we will use as missing data \( n \) \( z_i \) variables, one per site, with distribution \( P\{z_i = s\} = \alpha_s \), that is the probability that the \( s \)th Gamma mode is selected at site \( i \).

The EM algorithm alternates between two steps: the E-step computes the condition expectation of the logarithm of the complete a posteriori probability function, with respect to the missing variables, based on the actual parameter value. The M-step updates the values of the parameters, by maximizing the expression obtained in the E-step with respect to each parameter on turn, i.e.,

E-step: \( Q(\theta, \theta') = E\{\log p(y, z \mid \theta) \mid y, \theta'\} \) \tag{10}

M-step: \( \theta^{t+1} = \arg \max_{\theta} Q(\theta, \theta') \) \tag{11}

Denoting

\[ w_{si} = E\{P(z_i = s) \mid y, \theta'\} \tag{12} \]

and taking into account that \( \sum \alpha_s = 1 \), then \( \theta^{t+1} \) can be found among the stationary points of the Lagrangean
\[ L = \sum_{i=1}^{N} \sum_{s=1}^{K} (\log(\lambda_s) - \log(\Gamma(\alpha_s))) + \log(y_i^{\alpha_s-1}) - \lambda_s \times y_i + \log(\alpha_s) \times w_{is}^t + \lambda \times (\sum_{s=1}^{K} \alpha_s - 1) \] (13)

where \( \lambda \) denotes a Lagrange multiplier. The expression for \( w_{is}^t \) (see [10]) is given by

\[ w_{is}^t = \frac{\alpha_s \times f(y_i \mid \theta_i^t)}{\sum_{r=1}^{K} \alpha_r \times f(y_i \mid \theta_r^t)} \] (14)

In the M-step, after differentiating \( L \) in order to the unknown parameters and setting the derivatives to zero, we obtain a closed solution for the updating of the a priori probabilities \( \alpha_i \)'s, but numerical iteration is needed for determining parameters \( a_i \)'s and \( \lambda_i \)'s of the Gamma densities. Expression (15) gives the update expression for \( \alpha_i \)'s.

\[ \alpha_{i}^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} w_{is}^t \] (15)

Equation (16) and (17) give the update expressions for the parameters \( \lambda_i \)'s and \( a_i \)'s.

\[ \lambda_s = \frac{a_s \times \sum_{i=1}^{N} w_{si}}{\sum_{i=1}^{N} y_i \times w_{si}} \] (16)

\[ a_s = \Psi^{-1} \left[ \frac{\log(\lambda_s) \times \sum_{i=1}^{N} w_{si} + \sum_{i=1}^{N} \log(y_i) \times w_{si}}{\sum_{i=1}^{N} w_{si}} \right] \] (17)

with \( \Psi(a_s) = \frac{\Gamma'(a_s)}{\Gamma(a_s)} \) (18)

Expressions (17) and (18) are iteratively recomputed until convergence is obtained (few iterations are needed), starting from initial values computed from the observed data \( y \). The initial parameter values are calculated in such a way, that the initial probability function is a sum of equidistant Gammas that span the most representative data range.

3. GRAPH CUT ENERGY MINIMIZATION

As described in Section 1, we are concerned in maximizing expression (8), where the data model parameters have been substituted by the EM estimates.

This is equivalent to minimizing the logarithm of the expression, the total energy, so we may rewrite the problem in the following way:

\[ -\log p(x \mid y) \propto E(x_1, \ldots, x_n) = \sum_{i=1}^{n} E^i(x_i) + \sum_{i < j} E^{i,j}(x_i, x_j) \] (19)
with

\[
E_i^j(x_j) = -\log \left[ \sum_{j=1}^{N} \alpha_j \times f(y_j \mid x_j, \theta_j) \right] \tag{20}
\]

and

\[
E_{i,j} = -\beta \times \delta(x_i, x_j), \quad (i,j) \text{ being a 2-pixel-clique.} \tag{21}
\]

Here, the unary terms \(E_i(x_i)\) are derived from the observed data and measure the cost of assigning the label \(x_i\) to pixel \(i\). \(E_{i,j}(x_i, x_j)\) represents the pairwise interactions and measures the cost of assigning label \(x_i\) and \(x_j\) to pixels \(i\) and \(j\). The sum evaluates all adjacent pairs \((i,j)\), that build up a two-point-clique of a second order neighborhood (8 pixels). Labels \(x_i\) and \(x_j\) may take the values 1 and 2, corresponding to the oil and water class, respectively.

The energy function given in equation (19), is a sum of functions of up to two binary variables at a time, thus belonging to the \(F^2\) class described in [7]. Furthermore, each term \(E_{i,j}\) satisfies the inequality

\[
E_{i,j}^1(0,0) + E_{i,j}^1(1,1) \leq E_{i,j}^1(0,1) + E_{i,j}^1(1,0), \tag{22}
\]

so that \(E(x_1, \ldots, x_n)\) is a regular function and consequently graph-representable. In these circumstances, minimizing (19) corresponds exactly to computing the min-cut of a certain graph. The global minimum of (19) is exactly computed.

The characteristics of the energy function in (19) allow us to follow the general graph-cut construction method described in [7]. Because the minimum cut can efficiently be computed by max-flow algorithms, the resulting algorithm will have a low order polynomial running time on the number of pixels.

The directed graph will contain \(n + 2\) vertices: \(v = \{s,t,v_1, \ldots, v_n\}\), where \(s\) and \(t\) are terminal vertices. Each non-terminal vertex \(v_i\) encodes the binary variable \(x_i\).

The construction method is additive, i.e., we will construct a graph for each energy term in (19) and merge all the graphs together. We consider first the unary term \(E_i\), that depends only on one variable \(x_i\), and add an edge \((s,v_i)\) if \(E_i(1) < E_i(2)\), otherwise an edge \((v_i,t)\). The weight assigned to this edge is \(|E_i(1) - E_i(2)|\) and is represented in Figure 2, on the left side. The term \(E_{i,j}\), that depends on the two variables \(x_i\) and \(x_j\), will be represented by a graph containing three edges. These edges are represented in Figure 2, on the right side, where \(A, B, C\) and \(D\) have been defined as follow:

\[
A := E_{i,j}(1,1) = -\beta, \quad B := E_{i,j}(1,2) = 0, \quad C := E_{i,j}(2,1) = 0, \quad D := E_{i,j}(2,2) = -\beta
\]

After constructing graphs for all the energy terms and merging them, we obtain a graph like the one depicted in Figure 3.

![Fig. 2: Directed graphs representing unary and binary energy terms for labels \(x_i\) and \(x_j\). The left side stands for the term \(E_i\), when \(E_i(1) < E_i(2)\), see (a), and when \(E_i(2) \leq E_i(1)\), see (b). The right side stands for the term \(E_{i,j}\), see (c).](image-url)
It has nonnegative edge weights derived from the energy expression and s and t are, respectively, the sink and the source terminal vertices, representing the oil and the water classes.

![Directed graph](image)

**Fig 3**: Directed graph constructed from energy function for two-classes segmentation problem for \( \{x_1, \ldots, x_n\} \). Cuts represent labeling and cut costs represent energy.

A cut \( C = S, T \) on this graph is a partition of the vertices \( V \) into two disjoint sets \( S \) and \( T \) such that \( s \in S \) and \( t \in T \). The cost of the cut is defined as the sum of costs of all edges that go from \( S \) to \( T \).

The cut solution given by the algorithm defines a binary partition of the graph viewed as a binary-valued labeling. (see Fig.3), and the cost of this cut corresponds to the minimum of the energy function.

Regarding the choice of the \( \beta \) parameter, in a first phase we have tuned it manually. In a second phase, a pseudo-likelihood approach has been followed and \( \beta \) was determined using the Newton algorithm.

### 4. RESULTS: SIMULATED AND REAL IMAGES

In the following, results of applying the described segmentation algorithm are given for simulated and real images. The algorithm has three main steps:

1. the operator chooses an arbitrary number of regions of interest (ROI’s) per mouse-click for oil and for water.
2. EM parameter estimation of the class intensity probability distribution is computed.
3. a graph–cut algorithm is run for energy minimization.

Figure 5 shows a simulated SAR image of a fictitious oil spill. The image has been generated from the ground-truth given in Figure 4 using the class histograms of Figure 6.

![Simulated SAR image](image)

**Fig. 5**: Simulated SAR image

![Ground-truth](image)

**Fig. 4**: Ground-truth used for simulating an oil spill. Black represents oil and white water.
Fig. 6: Probability functions used to generate the simulated image with superimposed histogram of generated data set. A 3 modes function for water and 2 modes function for oil was used.

In this particular example, a ROI of 252 pixels for the oil class and of 297 pixels for water was used for parameter estimation. After 20 EM iterations, the algorithm gives the following probability distribution approximations (Figure 7):

Fig. 7: True and estimated class densities for water (on the left) and oil (on the right).

The result of applying the graph-cut segmentation tool using the estimated class distributions given above is depicted in Figure 8. No image enhancement method was further applied to the segmented image and an error rate of 0.7% was achieved.

Fig. 8: Output labeling of the segmentation tool. Error rate is circa 0.7%
The next image series shows the results of applying the segmentation tool to some real frames. Figure 9 is an ERS-1 frame of the 4th. October and documents an oil disaster occurred at that time in the Douro plume. It is an uncalibrated intensity image. The ROI’s used for parameter estimation are represented as colored squares.

Fig. 9: ERS-1 frame of Douro plume (on the left) and zoom of to dashed red square segment (on the right) with 5 ROI’s used for parameter estimation.

Fig. 10: Output labeling of the segmentation tool for the oil spill of the right upper corner of ERS-1 frame in Figure 9.

Fig. 11: Zoom of to dashed blue square segment of ERS-1 frame on Figure9 with 3 ROI’s used for parameter estimation (on the left). Estimated class probability distribution, with superimposed ROI’s histogram (on the right).
Finally the result of applying the algorithm to the probably best known example of an oil spill in the last years is shown in Figure 13: the Envisat frame of 17\textsuperscript{th} November 2002 in Galicia, corresponding to the Prestige disaster. Figure 13 shows the oil spill and the ROI’s used for parameter EM estimation and the image series of Figure 14 show the segmented parts.

5. CONCLUSIONS AND FURTHER WORK

The results obtained by applying the proposed segmentation tool to the simulated images have been very encouraging. The error rates are very low, circa 0.7\% to 2\%, even for fairly superimposed class histograms. A positive indication that Gamma mixtures of 2 to 3 modes per class give good EM estimates for real data has been found. The combination of a Gamma Mixture Model and a 2nd-order MRF seems to be adequate for the segmentation of an oil spill in SAR images.

The Graph Cut algorithm also proved to be a very efficient tool in the energy minimization step. Comparing the use of this approach to other energy minimization techniques, like ICM (not shown in this article), the graph cut technique is much faster and gives better segmentation results.

Our next steps will include the fully automatisation of the algorithm by developing unsupervised determination of mode number per class. We are also extending the algorithm for more than two classes, what involves applying the graph-cut step more than once.
REFERENCES