PolSAR Data Segmentation by Combining Tensor Space Cluster Analysis and Markovian Framework

Yinghua Wang, Chongzhao Han, Florence Tupin

***Abstract*—We present a new segmentation method for the fully polarimetric synthetic aperture radar (PolSAR) data by coupling the cluster analysis in the tensor space and the Markov random field (MRF) framework. The PolSAR data are usually obtained as a set of 3×3 Hermitian positive definite polarimetric covariance matrices, which do not form a Euclidean space. If we regard each matrix as a tensor, the PolSAR data space can be represented as a Riemannian manifold. Firstly, the mean shift algorithm is extended to the manifold to cluster such tensors. Then under the MRF framework, the data energy term is defined by the memberships of all tensors in all the clusters, while the smoothness energy term is defined according to the cluster overlap rates. These parameters regarding the cluster analysis are computed under the Riemannian framework. The total energy is minimized using a graph cut based optimization to achieve the segmentation results. The effectiveness of the proposed method is verified using real fully PolSAR data and synthetic images.**

***Index Terms*—Polarimetric synthetic aperture radar (PolSAR), image segmentation, Markov random field (MRF), cluster analysis, Riemannian manifold.**

I. Introduction

The segmentation of fully polarimetric synthetic aperture radar (PolSAR) data is still a challenging problem due to speckle. The speckle effects on the segmentation results can be suppressed to some extent when contextual information is introduced. Markov random fields (MRFs) [1] have proven to be powerful tools to impose spatial regularity constraint on the segmentation. Let $\mathcal{T}=\{\boldsymbol{T}_s, s\in\mathcal{S}\}$ be the observed data set, where $\mathcal{S}$ is the pixel set. The goal of segmentation is to obtain a label set $\mathcal{L}=\{L_s, s\in\mathcal{S}\}$. The MRF based segmentation can be achieved via minimizing an energy of the form

$$E(\mathcal{L})=\sum_{s\in\mathcal{S}} D_s(L_s)+\sum_{\{s,t\}\in\mathcal{N}} V(L_s,L_t) \qquad (1)$$

where $\{s,t\}\in\mathcal{N}$ indicates pixels s and t are neighbors. In (1), the *data term* $D_s(L_s)$ measures the disagreement between the label value L_s and the observation $\boldsymbol{T}_s$. The *smoothness term* $V(L_s,L_t)$ imposes the spatial regularity constraint and penalizes the spatial inhomogeneity in the label set.

Manuscript received Date; revised Date. This work was supported in part by a grant from the State Key Program for Basic Research of China (973) (No. 2007CB311006).

Y.-H. Wang and C.-Z. Han are with the Institute of Integrated Automation, Xi'an Jiaotong University, Xi'an 710049, China e-mail: yinghua.wang@stu.xjtu.edu.cn, dodgerbluester@gmail.com

F. Tupin is with the Institute TELECOM, TELECOM ParisTech, LTCI CNRS, 46, rue Barrault 75013 Paris, France e-mail: florence.tupin@telecom-paristech.fr

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2007.903065

$D_s(L_s)$ is commonly defined by the log-likelihood as $D_s(L_s)=-\ln(P(\boldsymbol{T}_s|L_s))$. The parametric models assume the observation distribution $P(\cdot)$ is known and defined by some parameters. The non-parametric models, such as the approach in [2] using the kernel methods to estimate $P(\cdot)$, do not impose any observation distributions. Thus they may be more appropriate for describing the complicated and arbitrary observations. For $V(L_s,L_t)$, the Potts model [3] is often used for classification, which is defined as $V(L_s,L_t)=\lambda U(L_s\neq L_t)$. The user defined parameter λ specifies the overall homogenization; $U(\cdot)$ is 1 if its argument is true and otherwise 0. This model assigns the same penalty for the label pair (L_s,L_t) as long as $L_s\neq L_t$. Thus it may eliminate some image details.

In this letter, we aim to define the data term by the non-parametric methods. However, some non-parametric methods such as the Parsen window techniques [4] often require to compute the Euclidean distances among the observations. Whereas the PolSAR data do not form a Euclidean space when for each pixel $s\in\mathcal{S}$, the observed value $\boldsymbol{T}_s$ is the 3×3 Hermitian positive definite (HPD) polarimetric covariance matrix. One way to extend these non-parametric methods to the PolSAR data space is to transform the HPD matrix space into a Euclidean space. In addition, we define the smoothness term using an adaptive $V(L_s,L_t)$ which is dependent on the specific (L_s,L_t) values. The cluster overlap rate (OLR) [5] is extended to the matrix space for defining $V(L_s,L_t)$.

The contribution of this paper is twofold: 1) we represent the PolSAR data space as a Riemannian manifold, then 2) we propose a practical segmentation algorithm. Firstly, the clustering of PolSAR data over the manifold is achieved by extending the original mean shift algorithm [6]. Then a new MRF based segmentation method employing the cluster analysis is presented. Under the Riemannian framework, we compute the degrees of membership of all polarimetric covariance matrices in all the clusters and the OLRs between every two clusters, which are respectively used to define the data term and the smoothness term in the MRF framework. The total energy is minimized using a graph cut based optimization to achieve the regularized segmentation.

II. PolSAR Data Space

The PolSAR data space can be represented as a Riemannian manifold as follows. A Riemannian manifold $\mathcal{M}$ is a differentiable manifold endowed with a Riemannian metric. The Riemannian metric smoothly assigns to each $\boldsymbol{X}\in\mathcal{M}$ an inner product on the tangent space $T_{\boldsymbol{X}}$. The space of $n\times n$

HPD matrices is known to be a differentiable manifold [7]. We endow it with the Riemannian metric defined in [8], where the inner product between two tangent vectors $\boldsymbol{y}, \boldsymbol{z} \in T_{\boldsymbol{X}}$ is given by

$$< \boldsymbol{y}, \boldsymbol{z} >_{\boldsymbol{X}} = \text{trace}\left(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{y}\boldsymbol{X}^{-1}\boldsymbol{z}\boldsymbol{X}^{-\frac{1}{2}}\right). \quad (2)$$

Note that (2) was initially defined for the space of symmetric positive definite (SPD) matrices. In this paper, we consider the straightforward extension of this framework to define the Riemannian manifold for the $n \times n$ HPD matrices, and in particular, the PolSAR data. Indeed all the concepts of [8] can be extended since the exponential, logarithm and power are well-defined for HPD matrices [9] as well.[1] Henceforth each HPD matrix on $\mathcal{M}$ will also be termed as a *tensor*, the same terminology used in [8]. Following [8] we now briefly describe some key concepts that will be used in the sequel.

For each point $\boldsymbol{X} \in \mathcal{M}$, two maps are defined in a neighborhood of $\boldsymbol{X}$ between $\mathcal{M}$ and $T_{\boldsymbol{X}}$. The *exponential map* $exp_{\boldsymbol{X}} : T_{\boldsymbol{X}} \rightarrow \mathcal{M}$ maps each tangent vector $\boldsymbol{y} \in T_{\boldsymbol{X}}$ to the point $\boldsymbol{Y} \in \mathcal{M}$. The inverse of the exponential map at point $\boldsymbol{X}$ is the *logarithm map* $log_{\boldsymbol{X}} : \mathcal{M} \rightarrow T_{\boldsymbol{X}}$, which maps each point $\boldsymbol{Y} \in \mathcal{M}$ to the tangent vector $\boldsymbol{y} \in T_{\boldsymbol{X}}$. The exponential and logarithm maps associated to (2) are defined as

$$exp_{\boldsymbol{X}}(\boldsymbol{y}) = \boldsymbol{X}^{\frac{1}{2}} \exp\left(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{y}\boldsymbol{X}^{-\frac{1}{2}}\right)\boldsymbol{X}^{\frac{1}{2}} \quad (3)$$

$$log_{\boldsymbol{X}}(\boldsymbol{Y}) = \boldsymbol{X}^{\frac{1}{2}} \log\left(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{X}^{-\frac{1}{2}}\right)\boldsymbol{X}^{\frac{1}{2}} \quad (4)$$

where $\exp(\cdot)$ and $\log(\cdot)$ are the matrix exponential and logarithm, respectively.

Here each tangent vector $\boldsymbol{y}$ is an $n \times n$ Hermitian matrix. Since there are only n^2 independent real coefficients in the upper or lower triangular part of $\boldsymbol{y}$, the minimal representation for $\boldsymbol{y}$ in $T_{\boldsymbol{X}}$ is given by the vector operator

$$\text{Vec}_{\boldsymbol{X}}(\boldsymbol{y}) = \text{Vec}_{\boldsymbol{I}}(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{y}\boldsymbol{X}^{-\frac{1}{2}}) \quad (5)$$

where $\boldsymbol{I}$ is the identity matrix, and $\text{Vec}_{\boldsymbol{I}}(\cdot)$ is defined as

$$\begin{aligned}\text{Vec}_{\boldsymbol{I}}(\boldsymbol{y}) =& [y_{1,1}, \sqrt{2}\text{Re}(y_{1,2}), \sqrt{2}\text{Im}(y_{1,2}), \sqrt{2}\text{Re}(y_{1,3}),\\ & \sqrt{2}\text{Im}(y_{1,3}), ..., y_{2,2}, \sqrt{2}\text{Re}(y_{2,3}), ..., y_{n,n}]^T.\end{aligned} \quad (6)$$

In (6), $(\cdot)^T$ means transpose; $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ take the real and imaginary part of a complex number, respectively. The mapping $\text{Vec}_{\boldsymbol{X}}(\cdot)$ realizes an isomorphism between $T_{\boldsymbol{X}}$ and $\mathbf{R}^{n^2}$ with the canonical metric.

The geodesic length between two tensors $\boldsymbol{X}$ and $\boldsymbol{Y}$ defines the *Riemannian distance* $d(\boldsymbol{X}, \boldsymbol{Y})$, which can be computed by

$$d^2(\boldsymbol{X}, \boldsymbol{Y}) = \text{trace}\left(\log^2\left(\boldsymbol{X}^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{X}^{-\frac{1}{2}}\right)\right). \quad (7)$$

The mean value $\bar{\mu}$ of a set of tensors $\{\boldsymbol{X}_i\}_{i=1}^n$ on $\mathcal{M}$ is defined as the tensor on $\mathcal{M}$ that minimizes the sum of the squared distances: $\bar{\mu} = \arg\min_{\boldsymbol{X}\in\mathcal{M}} \sum_{i=1}^n d^2(\boldsymbol{X}, \boldsymbol{X}_i)$. It can be computed by the gradient descent algorithm as follows:

$$\bar{\mu}_{t+1} = exp_{\bar{\mu}_t}\left(\frac{1}{n}\sum_{i=1}^n log_{\bar{\mu}_t}(\boldsymbol{X}_i)\right). \quad (8)$$

The linear interpolation $\boldsymbol{Z}$ of two tensors $\boldsymbol{X}$ and $\boldsymbol{Y}$ is on the geodesic joining these two tensors. It can be computed by

$$\boldsymbol{Z} = exp_{\boldsymbol{X}}(t \cdot log_{\boldsymbol{X}}(\boldsymbol{Y})), \qquad t \in [0, 1]. \quad (9)$$

[1]Consider a Hermitian matrix $\boldsymbol{W}$. Let $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{*T}$ be the eigenvalue decomposition of $\boldsymbol{W}$, where $(\cdot)^{*T}$ denotes the Hermitian transpose. The exponential of $\boldsymbol{W}$ is given by $\exp(\boldsymbol{W}) = \boldsymbol{U}\exp(\boldsymbol{D})\boldsymbol{U}^{*T}$, where $\exp(\boldsymbol{D})$ is the diagonal matrix of the eigenvalue exponentials. For a HPD matrix $\boldsymbol{W}$, its logarithm and power can be computed similarly, by replacing the eigenvalue exponentials with the eigenvalue natural logarithms and powers, respectively.

TABLE I
THE MS CLUSTERING ALGORITHM FOR POLSAR DATA

Input: $\boldsymbol{T}_s$, $s \in \mathcal{S}$, h, and M
- For each pixel s
  - Map $\boldsymbol{T}_s$ into the tangent space by $\boldsymbol{t}_s = log_{\boldsymbol{I}}(\boldsymbol{T}_s)$
  - Obtain the minimal representation $\boldsymbol{x}_s$ for $\boldsymbol{t}_s$ by $\boldsymbol{x}_s = \text{Vec}_{\boldsymbol{I}}(\boldsymbol{t}_s)$
  - Compute the convergence point $\boldsymbol{y}_s$ corresponding to $\boldsymbol{x}_s$ using (13)
  - Change $\boldsymbol{y}_s$ back to the 3×3 tangent vector $\boldsymbol{z}_s$ by $\boldsymbol{z}_s = \text{Vec}_{\boldsymbol{I}}^{-1}(\boldsymbol{y}_s)$
  - Map $\boldsymbol{z}_s$ back to the tensor space by $\boldsymbol{Z}_s = exp_{\boldsymbol{I}}(\boldsymbol{z}_s)$
- As proposed in [6], obtain the clustering results by grouping the pixels whose $\boldsymbol{y}_s$ values are closer than h into the same cluster
- Eliminate the clusters containing less than M pixels, which results in N_c clusters $\{\mathcal{S}_p\}_{p=1}^{N_c}$
- For each pixel s, assign $L_s = p$ if $\boldsymbol{y}_s \in \mathcal{S}_p$

Output: L_s, $\boldsymbol{Z}_s$, $s \in \mathcal{S}$, and N_c

III. MEAN SHIFT (MS) CLUSTERING OF POLSAR DATA

We first briefly review the MS algorithm [6], to which the readers may refer for more details. The MS algorithm extracts the local maxima of the empirical probability density function (p.d.f.), i.e., so-called density modes, in the feature space.

Given n data points $\{\boldsymbol{x}_i\}_{i=1}^n$ in the d dimensional Euclidean space $\mathbf{R}^d$, the kernel density estimator at point $\boldsymbol{x}$ with kernel $K(\boldsymbol{x})$ and bandwidth h is given by

$$\hat{f}_{h,K}(\boldsymbol{x}) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^n k\left(\left\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right\|^2\right) \quad (10)$$

where $k(x)$ is the profile of the kernel $K(\boldsymbol{x})$ and $c_{k,d}$ is the normalization constant. The gradient of $\hat{f}_{h,K}(\boldsymbol{x})$ is

$$\bigtriangledown\hat{f}_{h,K}(\boldsymbol{x}) = \hat{f}_{h,G}(\boldsymbol{x})\frac{2c_{k,d}}{h^2 c_{g,d}}m_{h,G}(\boldsymbol{x}) \quad (11)$$

where G is another kernel and its profile $g(x) = -k'(x)$; $m_{h,G}(\boldsymbol{x})$ is the MS vector, expressed as

$$m_{h,G}(\boldsymbol{x}) = \frac{\sum_{i=1}^n \boldsymbol{x}_i g(\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\|^2)}{\sum_{i=1}^n g(\|\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\|^2)} - \boldsymbol{x}. \quad (12)$$

For each data point $\boldsymbol{x}$, the general MS procedure contains two steps:

- Initialize $\boldsymbol{y}_0$ with $\boldsymbol{y}_0 = \boldsymbol{x}$.
- Update $\boldsymbol{y}_j$ by $\boldsymbol{y}_{j+1}$ until $\|\boldsymbol{y}_{j+1} - \boldsymbol{y}_j\|$ is small enough, where

$$\boldsymbol{y}_{j+1} = \boldsymbol{y}_j + m_{h,G}(\boldsymbol{y}_j). \quad (13)$$

Then we apply the MS algorithm to the PolSAR data. Since this algorithm is defined for the Euclidean space, we will transform first the PolSAR data space into a vector space. This transformation can be achieved by the logarithm map $log_{\boldsymbol{I}}(\cdot)$

and the vector operator $\mathrm{Vec}_{\boldsymbol{I}}(\cdot)$, which is equivalent to using the Log-Euclidean metric [10].

Table I depicts the MS clustering algorithm for PolSAR data. The input includes the polarimetric covariance matrix $\boldsymbol{T}_s$ for each pixel s, the MS bandwidth h, and the defined minimum number of pixels M for each cluster; the output contains each pixel's label value L_s and density mode $\boldsymbol{Z}_s$ that $\boldsymbol{T}_s$ belongs to, and the number of clusters N_c.

IV. MRF Based Segmentation Algorithm

The MRF based segmentation is achieved via minimizing the energy in (1), in which $D_s(L_s)$ and $V(L_s, L_t)$ will be newly defined in the following.

It is known that the fuzzy memberships in the Fuzzy k-means algorithm [4] describe the agreement between each observed value and each cluster. Therefore we define the data term by the memberships of all tensors in all the clusters.

Different from [4], we compute the memberships using the results of the MS algorithm in Table I, and the computations are carried out under the Riemannian framework. The membership depends on the cluster center. Once L_s and $\boldsymbol{Z}_s$, $s \in \mathcal{S}$, are obtained, the cluster centers $\{\boldsymbol{C}_p\}_{p=1}^{N_c}$ are given by

$$\boldsymbol{C}_p = \mathrm{E}_{\mathcal{M}}[\boldsymbol{Z}_s | L_s = p] \tag{14}$$

where $\mathrm{E}_{\mathcal{M}}[\cdot]$ denotes the mean of tensors computed by (8). The membership of $\boldsymbol{T}_s$ in the pth cluster, denoted by $u_{s,p}$, is based on the Riemannian distance (7):

$$u_{s,p} = \frac{1/d^2(\boldsymbol{T}_s, \boldsymbol{C}_p)}{\sum_{k=1}^{N_c} 1/d^2(\boldsymbol{T}_s, \boldsymbol{C}_k)}. \tag{15}$$

Equation (15) will be calculated for each pixel in each cluster. Then the data term $D_s(L_s)$ is given by

$$D_s(L_s = p) = -u_{s,p}. \tag{16}$$

Note that (16) does not impose any assumptions about the observation distribution. Moreover, N_c is not required to be set beforehand; it is derived from the parameters h and M in the MS algorithm, which can be set more intuitively.

Now we define the smoothness term $V(L_s, L_t)$ based on the cluster OLRs in the tensor space.

Given the cluster centers $\{\boldsymbol{C}_p\}_{p=1}^{N_c}$, the OLR between the pth and qth clusters, denoted by $\mathrm{OLR}(p, q)$, is defined as

$$\mathrm{OLR}(p, q) = \frac{P(\mathrm{saddle})}{\min[P(\boldsymbol{C}_p), P(\boldsymbol{C}_q)]} \tag{17}$$

where $P(\boldsymbol{C}_p)$ and $P(\boldsymbol{C}_q)$ are the p.d.f. values of the cluster centers $\boldsymbol{C}_p$ and $\boldsymbol{C}_q$, respectively; $P(\mathrm{saddle})$ is the p.d.f. value of the saddle point on the geodesic joining these two cluster centers. Equation (17) has the same expression as the OLR definition in [5], but all the p.d.f. values are estimated in a different way. If we extend the kernel density estimator (10) to the manifold using the Riemannian distance (7), the p.d.f. value of a tensor $\boldsymbol{X}_s$ with profile k and bandwidth h is

$$P(\boldsymbol{X}_s) = \frac{c_k}{n} \sum_{t \in \mathcal{S}} k\left(\frac{d^2(\boldsymbol{X}_s, \boldsymbol{T}_t)}{h^2}\right) \tag{18}$$

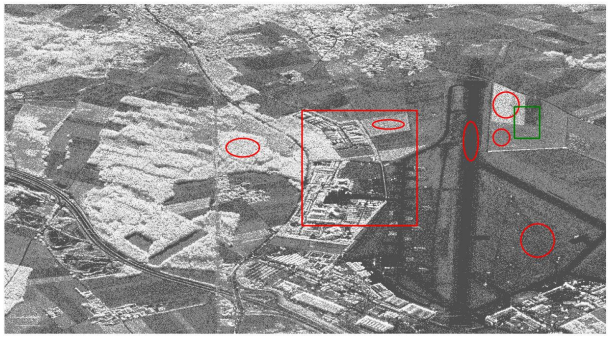


Fig. 1. The span image of the original data set. The two test sites used are chosen as the areas in the two boxes. The data contained in the six ellipses are employed to define the six classes in our simulated images.

where c_k is the normalization constant. The saddle point is the point with the lowest p.d.f. value on the geodesic that joins $\boldsymbol{C}_p$ and $\boldsymbol{C}_q$. This implies

$$P(\mathrm{saddle}) = \min_{\boldsymbol{X}_s} P(\boldsymbol{X}_s) \tag{19}$$

where each point $\boldsymbol{X}_s$ on the geodesic can be computed by (9). Finally, with $\{\mathrm{OLR}(p, q)\}_{(p,q) \in \{1,2,\ldots,N_c\}^2}$, we have the definition for $V(L_s, L_t)$:

$$V(L_s, L_t) = \begin{cases} 0 & L_s = L_t \\ \lambda \frac{\mathrm{OLR}(L_s, L_t)}{\max\limits_{(p,q)} \mathrm{OLR}(p,q)} & L_s \neq L_t \end{cases} \tag{20}$$

which is the same as the Potts model if for all the $(p, q) \in \{1, 2, ..., N_c\}^2$, we assign $\mathrm{OLR}(p, q) = 1$.

Notice that (20) provides adaptive smoothness depending on the class labels. When the pth and qth clusters are well separated in the feature space, $\mathrm{OLR}(p, q)$ tends to be 0. No regularization is imposed in this case. Thus edges between the pth and qth classes are preserved. On the contrary, if the pth and qth clusters are totally overlapping, $\mathrm{OLR}(p, q)$ becomes 1. In this case the spatial relations are essential and our smoothness term induces strong regularization.

The total energy (1) is minimized using a graph cut based optimization, i.e., the α-β-swap algorithm [11], which is a fast algorithm converging to the approximate global minimum with guaranteed optimality bounds.

V. Quantitative Evaluation of Segmentations

The quantitative segmentation evaluation is fulfilled in the same way as [12]. Given a set of segments $\{\mathcal{Q}_i\}_{i=1}^{N_c}$ and a ground truth region $\mathcal{G}$, the ratio $R_{\mathrm{IU}}(\mathcal{G}) \in [0, 1]$ is defined as

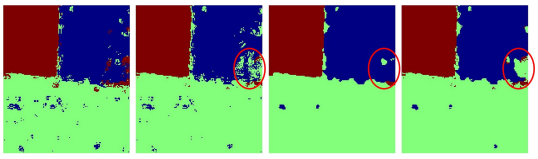
$$R_{\mathrm{IU}}(\mathcal{G}) = \max_i \frac{|\mathcal{Q}_i \bigcap \mathcal{G}|}{|\mathcal{Q}_i \bigcup \mathcal{G}|}. \tag{21}$$

$R_{\mathrm{IU}}(\mathcal{G})$, based on the cardinality of set intersection and union, measures the best spatial support for the region $\mathcal{G}$. For an entire data set, the segmentation performance can thus be evaluated by computing the mean R_{IU} value, denoted by $\bar{R}_{\mathrm{IU}}$, across all the ground truth regions.

VI. Experimental Results

The German Aerospace Center (DLR) E-SAR L-band fully PolSAR data[2] are used for experiments. The original images

[2]These data are downloaded from http://earth.esa.int/polsarpro/datasets.html.



(a) (b) (c) (d)

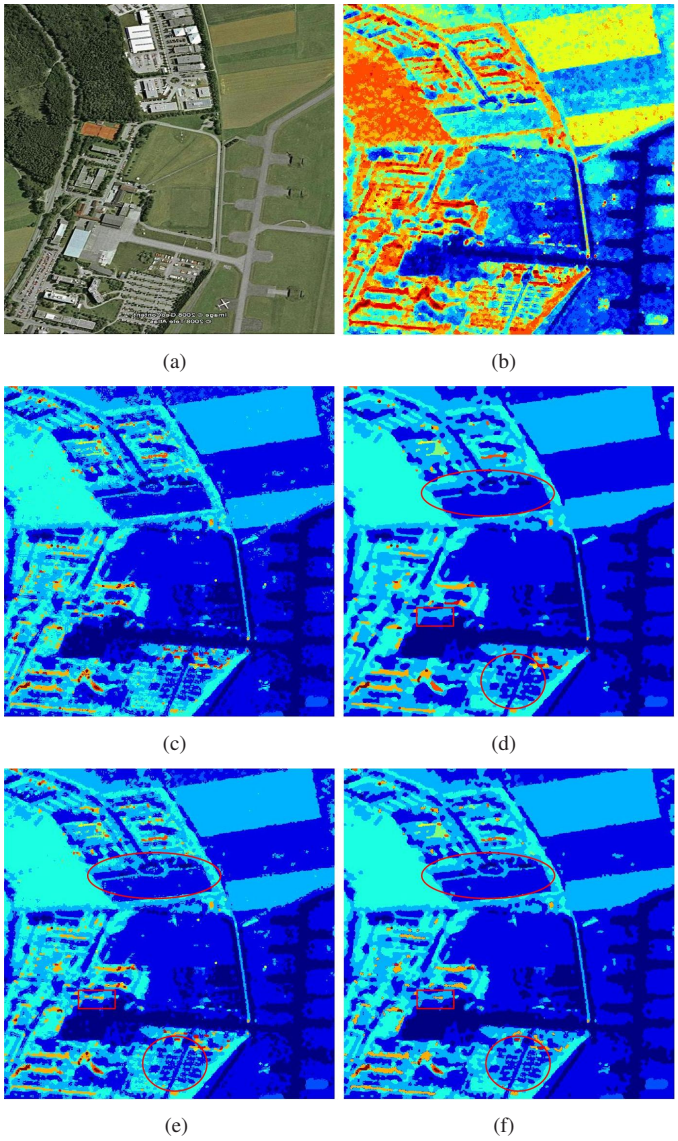
Fig. 2. Segmentation of the test site in the green rectangle in Fig. 1 into $N_c = 3$ classes using (a) WishAdaptNum, (b) Mem+0, (c) Mem+Potts with $\lambda = 0.05$, and (d) Mem+OLR with $\lambda = 0.05$.

have 1540×2816 pixels. The span image is shown in Fig. 1. Two sites in Fig. 1 are tested first, then a simulated data set constructed from this E-SAR data set is tested.

The proposed MRF based approach is denoted by Mem+OLR (using memberships to define the data term and OLRs to define the smoothness term). Firstly, to verify the effectiveness of the proposed data term, we assign the smoothness term of Mem+OLR to be 0 (no regularization is imposed) to get another approach Mem+0. Mem+0 is compared with the Wishart clustering method having adaptive number of clusters (denoted by WishAdaptNum) [13]. This is equivalent to comparing the proposed data term with the Wishart log-likelihood (WLL). Then, to demonstrate the effectiveness of the proposed smoothness term, Mem+OLR is compared with Mem+0 and Mem+Potts (substituting the smoothness term by the Potts model). In our experiments, speckle filtering is implemented beforehand using Lee's filter [14]. All the kernels used are the uniform kernels. In each segmentation, the classes are rank ordered by span and colored from dark blue to green to dark red according to their rank.

The first test site is selected as the 176×152 image in the green rectangle in Fig. 1. Since it is obviously composed of 3 classes, it can help us to choose the appropriate values for the MS parameters h and M. $h = 0.5$, 1, 1.5, 2 and $M = 20$, 40, 60 are tested. With $h = 1$ and $M = 40$, the image is rightly segmented into 3 classes by our methods. The results are shown in Fig. 2. Let us first consider the case without regularization. Fig. 2(a) and (b) show respectively the WishAdaptNum result and the Mem+0 result. The similar segmentation performance here verifies the effectiveness of the proposed data term. More regular results can be observed after regularization, as shown in Fig. 2(c) and (d). Now the same data term is used, but the small structures are better preserved as we use the proposed smoothness term [Fig. 2(d)] rather than the Potts model [Fig. 2(c)], such as the region in the red ellipse. This verifies the performance improvement of the proposed smoothness term over the Potts model. For the following test sites (including the simulated images), h and M are fixed to 1 and 40.

The second test site is the 512×512 image in the red rectangle in Fig. 1. The corresponding optical image [Fig. 3(a)] shows that this test site is complex enough due to plenty of typical targets. The segmentation results are shown in Fig. 3(b)−(f). Let us first compare the Mem+0 result [Fig. 3(c)] with the WishAdaptNum result [Fig. 3(b)]. It is observed that Mem+0 provides a more detailed segmentation of strong scatterers, and WishAdaptNum more finely segments the low return areas. The regularized results by various methods are



(a) (b) (c) (d) (e) (f)

Fig. 3. (a) The optical image from Google Earth ©2008 Google for the test site in the red rectangle in Fig. 1. Segmentation of this test site into $N_c = 11$ classes using (b) WishAdaptNum, (c) Mem+0, (d) Mem+Potts with $\lambda = 0.05$, (e) Mem+OLR with $\lambda = 0.05$, and (f) Mem+OLR with $\lambda = 0.25$.

shown in Fig. 3(d)−(f), in which we obtain more uniform region for each class. Then we compare Mem+OLR with Mem+Potts. With the same λ value, although the Mem+OLR result in Fig. 3(e) is a little noisier than the Mem+Potts result in Fig. 3(d) due to weaker smoothness constraint for some classes, the details are better preserved by Mem+OLR, such as the building in the red rectangle and the fine structures in the red ellipses. When λ is set to 0.25 for Mem+OLR, the result in Fig. 3(f) shows comparable smoothness to the Mem+Potts result [Fig. 3(d)] while representing the fine structures as precisely as the Mem+OLR result with $\lambda = 0.05$ [Fig. 3(e)].

To evaluate the segmentation methods quantitatively, a simulated data set is built in the same way as [15]. A ground truth image with six class labels is first designed as shown in Fig. 4(a). Then we manually select six homogeneous regions from the real E-SAR data set as those enclosed in the red ellipses in Fig. 1 and use them to fill in the six classes. The span image of the simulated data is shown in Fig. 4(b). The segmentation

TABLE II
THE R_{IU} VALUES FOR THE SEGMENTATION RESULTS OF THE SIMULATED DATA SET

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	$\bar{R}_{\mathrm{IU}}$
WishAdaptNum	0.9028	0.8577	0.8945	0.8056	**0.9120**	0.9629	0.8892
Mem+0	0.9251	0.8792	0.8982	0.8203	**0.9061**	0.9691	0.8997
Mem+Potts, $\lambda = 0.1$	0.9523	0.9379	0.9390	0.9157	**0.9262**	0.9788	0.9416
Mem+Potts, $\lambda = 0.45$	0.9493	0.9125	0.9147	0.8858	**0.9148**	0.9671	0.9240
Mem+OLR, $\lambda = 0.1$	0.9211	0.9328	0.9323	0.8787	**0.9290**	0.9622	0.9260
Mem+OLR, $\lambda = 0.45$	0.8183	0.9128	0.9300	0.8914	**0.9362**	0.9734	0.9103

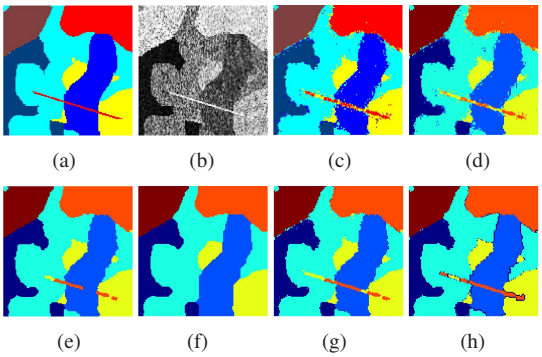


Fig. 4. (a) The ground truth label image. (b) The span image of the simulated data set. Segmentation of this data set into $N_c = 6$ classes using (c) WishAdaptNum, (d) Mem+0, (e) Mem+Potts with $\lambda = 0.1$, (f) Mem+Potts with $\lambda = 0.45$, (g) Mem+OLR with $\lambda = 0.1$, and (h) Mem+OLR with $\lambda = 0.45$.

results are shown in Fig. 4(c)−(h), and evaluated quantitatively in terms of the R_{IU} values, which are displayed in Table II. Firstly, $\bar{R}_{\mathrm{IU}}$ of Mem+0 is higher than WishAdaptNum, which verifies the performance improvement of the proposed data term over WLL. Then, Mem+OLR is compared with Mem+Potts. When $\lambda = 0.1$, $\bar{R}_{\mathrm{IU}}$ of Mem+OLR is lower than Mem+Potts due to weaker smoothness constraint for some classes, but for Class 5 R_{IU} of Mem+OLR is higher. Class 5 is the one in red color, to which the thin red line belongs. This means the fine structures are better preserved by the proposed smoothness term, which can also be observed by comparing Fig. 4(g) with (e). When $\lambda = 0.45$, for five of the six classes, using Mem+OLR yields higher R_{IU} values than Mem+Potts. This implies the proposed smoothness term performs better than the Potts model, as it is displayed in Fig. 4(h) and (f). To sum up, for the Potts model, more regular results may imply less details, yet with certain λ values, the proposed smoothness term can provide results with comparable smoothness while preserving most of the details.

VII. CONCLUSION AND PERSPECTIVE

We represent the PolSAR data space as a Riemannian manifold, and propose a new segmentation method applied to the data. The MS algorithm is first extended to cluster the data over the manifold. Then based on the clustering results, we propose the new definitions for the energy terms in the MRF framework. The effectiveness of the proposed method is verified by experimental results with the real fully PolSAR data and synthetic images. The proposed data term defined by cluster memberships shown to be advantageous over the Wishart log-likelihood. With the appropriate bandwidth for the MS algorithm, the proposed smoothness term defined by the cluster OLRs may yield better performance than the Potts model, in terms of preserving the image details. Moreover, no prior knowledge about the observation distribution or the cluster number is required in our method. The future work includes improving the proposed method using adaptive MS bandwidth and applying it to the terrain classification.

ACKNOWLEDGMENT

The authors would like to thank the ESA (http://earth.esa.int /polsarpro/datasets.html) for providing the data. They are also grateful to Dr. P. Fillard of INRIA Sophia Antipolis and the anonymous reviewers for their helpful suggestions.

REFERENCES

[1] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. London, U.K.: Springer-Verlag, 1995.
[2] C. He, G.-S. Xia, and H. Sun, "SAR images classification method based on Dempster-Shafer theory and kernel estimate," *Journal of Systems Engineering and Electronics*, vol. 18, no. 2, pp. 210–216, 2007.
[3] G.-S. Xia, C. He, and H. Sun, "A rapid and automatic MRF-based clustering method for SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 596–600, Oct. 2007.
[4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.
[5] M. Bouguessa, S. Wang, and H. Sun, "An objective approach to cluster validation," *Pattern Recognition Letters*, vol. 27, no. 13, pp. 1419–1430, Oct. 2006.
[6] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[7] R. Bhatia, *Positive Definite Matrices*. Princeton University Press, 2007.
[8] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 41–66, Jan. 2006.
[9] J. H. Gallier, *Geometric Methods and Applications: for Computer Science and Engineering*. New York: Springer-Verlag, 2001.
[10] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," *Magn. Reson. Med.*, vol. 56, no. 2, pp. 411–421, Jun. 2006.
[11] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[12] T. Malisiewicz and A. Efros, "Improving spatial support for objects via multiple segmentations," in *British Machine Vision Conference*, 2007.
[13] F. Cao, W. Hong, Y. Wu, and E. Pottier, "An unsupervised segmentation with an adaptive number of clusters using the $SPAN/H/\alpha/A$ space and the complex Wishart clustering for fully polarimetric SAR data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 11, pp. 3454–3467, Nov. 2007.
[14] J.-S. Lee, M. R. Grunes, and G. de Grandi, "Polarimetric SAR speckle filtering and its implication for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 5, pp. 2363–2373, Sep. 1999.
[15] Y. Yang, C.-Z. Han, and D.-Q. Han, "A Markov random field model-based fusion approach to segmentation of SAR and optical images," in *Proc. IGARSS*, 2008, pp. 802–805.