

---

# Spatial Sampling for Image Segmentation

MARIANO RIVERA<sup>1</sup>, OSCAR DALMAU<sup>1</sup>, WASHINGTON MIO<sup>2</sup> AND  
ALONSO RAMIREZ–MANZANARES<sup>3</sup>

<sup>1</sup>*Centro de Investigacion en Matematicas A.C, Guanajuato GTO 36000, Mexico*

<sup>2</sup>*Florida State University, Tallahassee FL 32306, USA*

<sup>3</sup>*Universidad de Guanajuato, Departamento de Matematicas, Guanajuato GTO 36000, Mexico*

*Email: mrivera@cimat.mx*

---

**We present a novel framework for image segmentation based on the Maximum Likelihood estimator. A common hypothesis for explaining the differences among image regions is that they are generated by sampling different Likelihood Functions called models. We adopt last hypothesis and, additionally, we assume that such samples come from independent and identically distributed random variables. Thus, the probability (likelihood) that a particular model generates the observed value (at a given pixel) is estimated by computing the likelihood of the sample composed with the surrounding pixels. This simple approach allows us to propose efficient segmentation methods able to deal with textured images. Our approach is naturally extended for combining different features. Experiments in interactive image segmentation, automatic stereo analysis, and denoising of brain water diffusion multi-tensor fields demonstrate the capabilities of our approach.**

*Keywords: Image segmentation, Early computer vision, Maximum likelihood estimation, Optical flow, Image denoising, Diffusion magnetic resonance imaging.*

*Received July 2010; revised February 2011*

---

## 1. INTRODUCTION

Image segmentation, an important issue in computer vision and image analysis, consists in partitioning an image into regions with similar characteristics: color, texture, local orientation, *etc.* Image segmentation is an ill-posed problem that is task and user dependent: a single image can be hand-segmented by different users in very different ways, even a same user can propose different segmentation by depending of the task he/she is accomplishing.

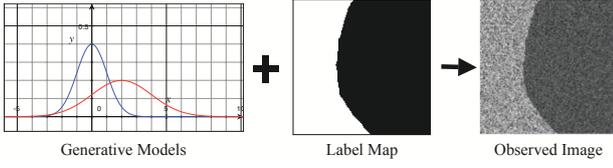
A powerful strategy for segmenting images is to consider the image lattice as a graph, and by means of spectral methods to compute a graph-cut [1, 2, 3, 4]. The main advantage of spectral clustering based methods is that they are modeless. It means that, given the pixel interactions, a partition of the image can be computed without previous knowledge either the number of clusters or the cluster's representatives (models). Spectral clustering algorithms can be seen as exploratory techniques that require a post-processing in order to obtain a desired solution.

On the other hand, the image segmentation is commonly presented as the solution of a combinatorial problem. Popular approaches to combinatorial image segmentation are based on Markov Random Field (MRF) models [5, 6]. A reason is their flexibility for being adapted to different circumstances, for example: color [7], motion and stereo disparity [8, 9]. In that order, max-flow graph-cut based techniques have successfully been applied for the direct label map computation [8, 9, 10, 11, 12]. A

different approach is to estimate the uncertainties on the label assignment (memberships) [13, 14, 15, 16, 17]. In the Bayesian framework, such memberships can be expressed in a natural way in terms of probabilities—leading to the named Probabilistic Segmentation (PS) methods.

In this paper we present a novel framework for probabilistic image segmentation. As illustrated in experiment section, the new framework can be used for solving many problems in image processing and computer vision. We present four applications: interactive image segmentation that integrates color and texture clues, stereo-disparity estimation, denoising and multitensor fields restoration. We select such applications because: first, they are active research topics; second, they are different enough for demonstrating the framework flexibility; and third, they allow us to illustrate different implementation details. In spite of the basic implementations presented here, they are competitive with methods of the state of the art specifically developed for such tasks.

In the following we introduce the notation used in this work. We assume that the observed image  $g : \Omega \rightarrow \mathbb{R}^n$  ( $n = 1$  for gray scale images and  $n = 3$  for color images) is generated by sampling unknown probability density functions named models  $\mathcal{M} = \{M_k\}_{k=1}^K$  with parameters  $\theta = \{\theta_k\}_{k=1}^K$ ;  $r$  and  $s$  denote pixel positions,  $\Omega$  is the set of all the pixels in a regular lattice and  $\mathcal{K} = \{1, 2, \dots, K\}$  the label set such that, the label field  $c : \Omega \rightarrow \mathcal{K}$  indicates the source for each pixel (see Figure 1). Then



**FIGURE 1.** Illustration of the generative model: Image regions are generated by independent samples of likelihood functions. In this example, the generative models are represented with Gaussian densities, blue and red curves

the task consists of solving the inverse problem: to segment the image  $g$  into  $K$  classes (*i.e.*, to estimate  $c$ ) and, in the case, the unknown parameters  $\theta$ . We denote by

$$v_k(r) = P(g(r)|\theta_k, c(r) = k). \quad (1)$$

the likelihood of observing  $g(r)$  given the model  $k$ . This can be seen as the preference of the data  $g(r)$  for the model  $k$ . Hence, the Maximum Likelihood (ML) estimator (classification) is given by

$$c^{ML}(r) = \underset{k}{\operatorname{argmax}} v_k(r). \quad (2)$$

This Winner Takes All (WTA) assignment is used as estimator of the true label,  $c(r)$ . A disadvantage of the ML estimator is its sensibility to noisy data that results in noisy segmentations. For improving the segmentation of noisy images one has two choices:

1. To increase the number of samples per pixel, *i.e.*, to acquire a set of independent observations  $\{g_i(r)\}_{i=1}^I$ . Then the sample noise contributions are averaged and therefore the ML estimator is improved. The likelihood of the sample is given by:

$$v_k(r) = \prod_i P(g_i(r)|\theta_k, c(r) = k). \quad (3)$$

2. To use prior knowledge that promotes smooth solutions. If this prior is coded as the probability  $P(c)$ , then, by using the Bayes' rule, the MAP estimator can be computed from the posterior probability:

$$c^{MAP} = \underset{c}{\operatorname{argmax}} \prod_r \prod_k v_k(r) P(c). \quad (4)$$

In any case, one can estimate, simultaneously, the segmentation and the model parameter. This joint estimation can be implemented by an EM strategy [18]: to estimate a segmentation (hard or probabilistic) by fixing the parameters and then to estimate the parameters by fixing the classification.

Note that both strategies, to acquire multiple samples per pixel or to use Bayesian regularization, are not mutually exclusive: the likelihood in (4) can be improved by using multiple samples. However, from last two options, Bayesian regularization is the preferred strategy given that, in general, we are limited to a single image [5, 6]. In such a

case, one can find broad literature of solution methods for the optimization problem stated in (4). Such techniques can be classified as combinatorial optimization approaches (the ones that try to directly estimate the  $c^{MAP}$ , a hard-segmentation) [1, 6, 10, 19, 20, 21, 22] and probabilistic approaches (the ones for estimating a hidden real vector field that represents the probability that  $c(r)$  takes a particular label, a PS) [13, 15, 23, 24].

In this work we present a method for improving the likelihood in lack of multiple observations. Our improved likelihoods can directly be used for computing a segmentation by means of the ML estimator (2) or can be used as prime matter for a Bayesian segmentation method that solve (4). The general idea is simple, we assume that all the pixels values are *i.i.d.* samples of generative models and the source is determined by the label map  $c$ . Since image regions are relative large (this assumption is frequently codified as a prior in Bayesian regularization), then the pixels in a small neighborhood are very likely samples of a unique model. Thus, the small pixel neighborhood can be assumed as multiple observations of the central pixel. Indeed, this simple idea is the underlying one of all spatial filtering techniques in image processing. Inspired on that, we propose a novel and efficient framework for image probabilistic segmentation. Our approach, differently from those frameworks that regularize (smooths) the pixel values, regularizes discrete likelihood densities.

We organize the paper as follows. The mathematical derivation of our strategy is presented in Section 2. Then, a summary of the algorithm is presented in Section 3. Next, applications that demonstrate the performance, flexibility and generalization capabilities of our framework are presented in Section 4. Finally, our conclusions are given in Section 5.

## 2. SPATIAL SAMPLING

### 2.1. Likelihood based on spatial samples

Let  $\mathcal{N}_r$  be a neighborhood of pixels centered at  $r$  (inclusive) and  $\mathcal{G}_r = \{g(s) : s \in \mathcal{N}_r\}$  their corresponding pixels values. For the moment, we assume the simple neighborhood  $\mathcal{N}_r = \{s : \|r - s\|^2 \leq \rho\}$ , where we denote the Euclidean norm by  $\|\cdot\|$ . In subsection 2.3 we will discuss details about the selection of the neighborhood.

Thus, we can assume  $\mathcal{G}_r$  a sample of the mixed likelihood:

$$\begin{aligned} P(\mathcal{G}_r|\theta, c; \pi) &= \sum_k \pi_k(r) \left[ \prod_{s \in \mathcal{N}_r} P(g(s)|\theta_k, c(s) = k) \right] \\ &= (\pi^T \tilde{v})(r); \end{aligned} \quad (5)$$

where we define the spatial likelihood

$$\tilde{v}_k(r) \stackrel{def}{=} \prod_{s \in \mathcal{N}_r} v_k(s) \quad (6)$$

and  $\pi(r) \in \mathbb{S}^K$  is a vector whose components are unknown mixture coefficients, where we denote by  $\mathbb{S}^K$  the simplex

with all the vectors with positive entries that sum one:

$$z \in \mathbb{S}^K \iff z \geq 0, \sum_k z_k = 1 \quad \text{for } k = 1, 2, \dots, K. \quad (7)$$

Thus,  $\pi_k(r)$  is the fraction of the sample  $\mathcal{G}_r$  generated with the  $k$ -th model.

Then, the image segmentation can be estimated from an estimator of  $\pi$  if an appropriated  $\mathcal{N}_r$  is selected, see subsection 2.3. Following we investigate two estimators of  $\pi$ :

1. Since (5) is a Linear Programming problem, then it is easy to prove that the ML estimator is the indicator vector:

$$p^{(1)}(r) = e^{k^*} \quad (8)$$

where  $e^k$  is the  $k$ -th basis vector and

$$k^* = \operatorname{argmax}_k \tilde{v}_k(r) = \operatorname{argmax}_k \hat{v}_k(r); \quad (9)$$

where the normalized spatially-computed likelihood,  $\hat{v}_k(r)$ , is given by

$$\hat{v}_k(r) = \frac{\tilde{v}_k(r)}{\sum_i \tilde{v}_i(r)}. \quad (10)$$

2. A soft estimation of  $\pi$  can be computed by the maximization:

$$p^{(2)}(r) = \operatorname{argmax}_\pi \frac{(\pi^T \hat{v})(r)}{\|\pi(r)\| \|\hat{v}(r)\|}. \quad (11)$$

That results in

$$p^{(2)}(r) = \hat{v}(r). \quad (12)$$

Therefore, in any of the last two cases, the estimation of  $\pi$  is reduced to the computation of  $\hat{v}_k(r)$ : spatial products of individual likelihoods, or sums of log-likelihoods:

$$\hat{v}_k(r) \propto \prod_{s \in \mathcal{N}_r} v_k(s) = \exp \left( \sum_{s \in \mathcal{N}_r} \log v_k(s) \right). \quad (13)$$

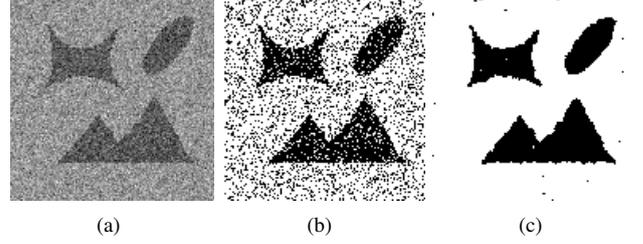
Finally, the hard segmentation (or pixels label field) is computed with

$$c^*(r) = \operatorname{argmax}_k \hat{v}_k(r) = \operatorname{argmax}_k \log \tilde{v}_k(r). \quad (14)$$

## 2.2. Multiple Sources: Combining multiple segmentations

Now we extend the above introduced probabilistic segmentation to the case of multiple sources, *i.e.*, to combine independent segmentation from different clues. After that, we will be in the capability of presenting the neighborhood choices and their algorithmic implications. First, we note that (13) can be written as

$$\hat{v}_k(r) \propto v_k(r) u_k(r) \quad (15)$$



**FIGURE 2.** ML segmentation ( $\operatorname{argmax}_k \hat{v}_k(r)$ ): (a) Binary image corrupted with Gaussian Noise [ $\mathcal{N}(m=0, \sigma=0.5)$ ], (b) segmentation based on the pixelwise likelihood  $v$  of each pixel  $r$  and (c) segmentation based on the likelihood  $u$  that is estimated with the neighbor pixels except  $r$ , see (16).

with

$$u_k(r) = \prod_{s \in \mathcal{N}_r \setminus \{r\}} v_k(s). \quad (16)$$

Eq. (15) can be understood as the combination of two independent sources: the likelihood estimated from the observed value,  $v_k(r)$ , and the likelihood,  $u_k(r)$ , estimated with the neighbor pixels except  $r$ , see Figure 2. We can give a further step by generalizing (15) to  $J$  independent sources and introducing their confidence factor:

$$\hat{v}_k(r) \propto \prod_{j=1}^J [v_k^{(j)}(r)]^{\alpha_j} = \exp \left( \sum_{j=1}^J \alpha_j \log v_k^{(j)}(r) \right); \quad (17)$$

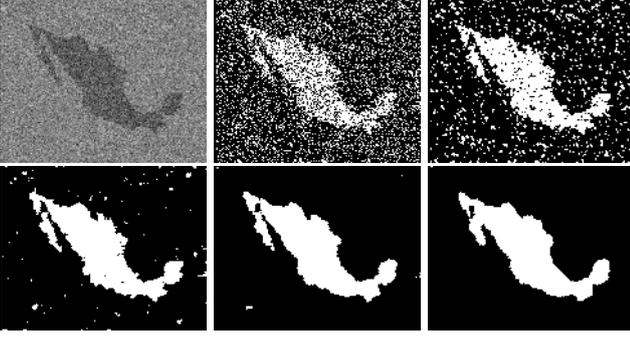
where  $\alpha_j$  is our grade of confidence of the  $j$ -th-source (the  $v^{(j)}$  likelihood) and it holds  $\alpha \in \mathbb{S}^J$ . So that  $\alpha_j = 1$  means that the  $j$ -th-source has the largest possible confidence and it becomes irrelevant as  $\alpha_j \rightarrow 0$ .

Eq. (17) is a simple form of combining a set of likelihoods (probabilistic segmentations). The different PS (sources) can result from the use of different clues as, for instance, color and local statistical descriptors for texture. We illustrate this capability in Section 4, Applications and Experiments.

## 2.3. On the Neighborhood Selection

We have shown that if the image to segment is composed of an assemble of relative large regions (*i.e.*, the label field  $c$  is “smooth”), then spatial samples (neighborhood of pixels) can alleviate the lack of multiple observations for each pixel. An accurate segmentation depends on selecting a pixel neighborhood such that its majority belongs to the correct class. Note that the right side of (14) defines a spatial filtering of the likelihoods in the log-space. Now we show that the proposed framework accepts both linear and nonlinear filters.

Following we present derivations of distinct log-likelihood filters by choosing a proper neighborhood and assuming that each pixel neighbor  $\mathcal{N}_r$  is an independent source for estimating the likelihood at the pixel of interest,  $r$ . We remark that the proposed PS algorithms can be constructed on well-known spatial filters, for example the



**FIGURE 3.** ML segmentation with a square neighborhood ( $\mathcal{N}_r = \{s : \|r-s\|_\infty \leq \rho\}$ ) for different values of  $\rho$ . The data corresponds to a binary Mexico map (top left) corrupted with Gaussian noise [ $N(m=0, \sigma=0.7)$ ] and the segmentations to  $\rho = 0, 1, 2, 3$  and  $4$ ; respectively.

ones in Refs. [25, 26, 27, 29, 30]. The novelty is the spatial filtering of log-likelihoods for segmentation purposes.

**Homogeneous Windows (HW).** The simplest neighborhood is a regular window centered at the pixel  $r$ :  $\mathcal{N}_r = \{s : \|r-s\|_m \leq \rho\}$ , where the parameter  $\rho$  controls the sample size and  $\|\cdot\|_m$  is a given metric. For instance, if the  $L_\infty$  norm is used, then the neighborhood is square-shaped and (13) is reduced to a box-filter in the logarithmic space of each likelihood layer [25]. The use of an homogeneous window is justified by the prior knowledge of relative large regions with smooth borders. Then, we can assume that, almost at any place, the proportion of the sample generated with the model of the central pixel is greater than the generated with the rest of the models. Figure 3 shows the segmentation based on the ML estimator for different values of  $\rho$ . Note that for large  $\rho$ -values the segmentation granularity is reduced and small details are over-smoothed.

**General Homogeneous Windows (GHW).** The use of the previous HW is equivalent to apply a box-shape linear filter in the log-domain. This result can be generalized to any arbitrary linear filter if each neighbor pixel is considered an independent source. Then, similarly to (17), the sources (neighbor pixels) are combined with a confidence factor that depends on their spatial distance to the central pixel  $r$ :

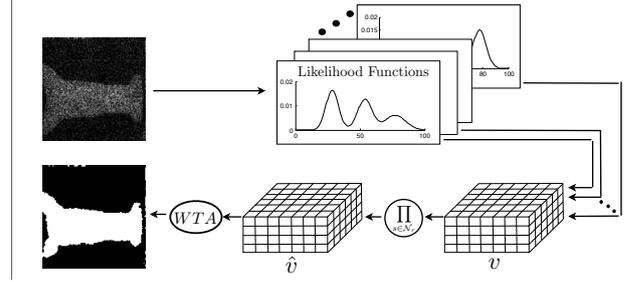
$$\hat{v}_k(r) \propto \prod_{s \in \mathcal{N}_r} [v_k(s)]^{\alpha(s)} = \exp \left( \sum_{s \in \mathcal{N}_r} \alpha(s) \log v_k(s) \right). \quad (18)$$

In particular, the Gaussian filtering results of choosing

$$\alpha(s) \propto \exp \left( -\frac{1}{2} (r-s)^T \Sigma^{-1} (r-s) \right), \quad (19)$$

where  $\Sigma$  is a covariance matrix. This can be understood as a variant weight that depends on the Mahalanobis spatial distance of the source (neighbor pixel) to the central pixel. Among GHW we can include the linear filters that result of an Homogeneous Diffusion [26] and a Membrane Potential [5, 27]. In particular, the homogeneous-membrane based segmentation,  $c$ , is computed by

$$c^*(r) = \operatorname{argmin}_k d_k^*(r) \quad \forall r$$



**FIGURE 4.** Illustration of MSL estimator for image segmentation. See text.

where  $d^* = \operatorname{argmin}_{d: \Omega \subset \mathcal{L} \rightarrow \mathbb{R}^K} U(d)$  is the minimizer of the layered membrane energy:

$$U(d) = \sum_{r \in \Omega} \left\{ \|d(r) - \log v(r)\|^2 + \frac{\lambda}{2} \sum_{s \in \mathcal{N}_r} \|d(r) - d(s)\|^2 \right\}, \quad (20)$$

where  $\Omega$  is the region of interest and  $\mathcal{L}$  is the lattice that corresponds to the pixels of the image. This optimization problem is equivalent to independently apply a linear filter to each layer  $\log v_k$ .

**Spatially Adapting Windows (SAW).** In order to improve the quality of the spatial sample, the neighborhood shape can be adapted by depending on the image local properties. For instance, an anisotropic filtering can be achieved by using the local structure tensor as an adaptive covariance matrix in (19):

$$\Sigma_{g,r}^{-1} = (H_\sigma * (\nabla g \nabla g^T))(r), \quad (21)$$

where  $H_\sigma$  is a Gaussian kernel with variance  $\sigma$  and  $*$  is the convolution operator [28]. We can make the confidence factors depending on the range value too. Then by combining spatial and range confidence factors we have a sort of bilateral filter in the log-likelihood space [29, 30]:

$$\begin{aligned} \hat{v}_k(r) &\propto \prod_{s \in \mathcal{N}_r} [v_k(s)]^{\alpha(s)\beta(s)} \\ &= \exp \left( \sum_{s \in \mathcal{N}_r} \alpha(s)\beta(s) \log v_k(s) \right); \end{aligned} \quad (22)$$

where

$$\beta(s) \propto \exp \left( -\frac{1}{2} [G_r - G_s]^T \Sigma_r^{-1} [G_r - G_s] \right) \quad (23)$$

is a confidence factor that depends on the range value and  $G_r = \mathbf{T}\{g(r)\}$  is a transformation of the range values. Instances of  $\mathbf{T}$  are the identity ( $G = g$ ) and the likelihood ( $G = v$ ).

It is not our aim to present an exhaustive filter list, but notice that the filter in the log-likelihood space is general and thus we are not constrained to use a particular filter kind. Such a log-likelihood filtering corresponds to a

**Algorithm 1** MSL

- 1: Given the Probability Likelihood functions (models) then:
- 2: Compute the individual likelihood  $v_k(r)$  for all model  $k$  and pixel  $r$ .
- 3: Compute the logarithm of the normalized likelihood:

$$d_k(r) = \log \left( v_k(r) / \sum_j v_j(r) \right), \quad \forall k, r.$$

- 4: Apply a spatial filtering,  $F\{\cdot\}$ , to each layer of the log likelihood:  $d_k \leftarrow F\{d_k\}$ .
- 5: Compute the spatial likelihood

$$\hat{v}_k(r) = \exp(d_k(r)) / \sum_j \exp(d_j(r)), \quad \forall k, r.$$

- 6: Compute the label map  $\tilde{c}(r) = \operatorname{argmax}_k \hat{v}_k(r), \quad \forall r$ .

particular neighborhood and confidence source factors of the likelihoods. As we will see in the developed applications (Section 4), even a simple GHW produce results of excellent quality.

### 3. MAXIMUM OF SPATIAL LIKELIHOODS (MSL) ALGORITHM

As summary of Section 2, we present the algorithm for estimating likelihoods based on multiple spatial samples. The procedure is defined in Algorithm 1 and is illustrated in Figure 4. Note that, the step 5 in Algorithm 1 can be obviated and the segmentation can directly be computed in the step 6 with  $\tilde{c}(r) = \operatorname{argmax}_k d_k(r), \quad \forall r$ .

Now, if we have feature vectors, the Algorithm 1 can be used for computing a set of likelihoods  $\{\hat{v}^{(j)}\}$ , each one corresponding to a feature. In such a case the final likelihood is computed by combining the individuals likelihoods by means of (17), see Figure 5.

### 4. APPLICATIONS AND EXPERIMENTS

The purpose of this section is to demonstrate the versatility of our proposal by implementing a variety of applications. Our main aim is to present a new viewpoint rather than a specific solution to each problem. Hence, we motivate the reader to pursuit further implementations by extending the here presented techniques or by incorporating this framework in the design of new algorithms. In order to guide the reader to put in practice our framework, we present the following applications: interactive image segmentation that integrates color and texture clues, stereo-disparity estimation, denoising and multi-tensor fields restoration.

We demonstrate by comparisons with methods of the state of the art that the use of the proposed framework leads one to better solutions; in particular, see Sections 4.1 and 4.4.

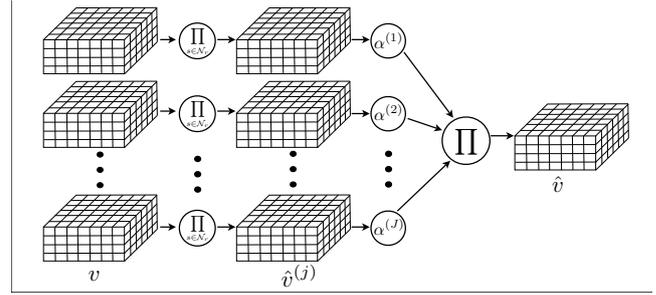


FIGURE 5. Combination of multiple probabilistic segmentations.

#### 4.1. Segmentation from multiples clues

Here, we develop an interactive procedure for multi-class image segmentation based on the MSL segmentation. The aim of the application is to demonstrate the importance of texture features for improving the quality of the segmentation. In order to present a direct comparison with methods of the state of the art, we opt for using a homogeneous filter (a GHW). However, we advise to the reader that the class edges location could be improved if a SAW scheme is used; see the presented applications in Sections 4.2, 4.3 and 4.4.

The models are empirically initialized from user's marked data (scribbles) on the image. As any interactive procedure, our method admits iterative scribble retouching but this possibility is not investigated here. The results computed with the initial scribbles are presented as the *final* segmentation. The purpose of this application is to demonstrate that the final segmentation is improved by combining multiples sources (likelihood vectors) and the source combination is naturally implemented in our proposal. Figure 5 illustrates the process.

We assume that the user scribbles define the multimap  $C : \mathcal{L} \rightarrow \{0\} \cup \mathcal{K}$  such that  $C(r) = k \in \mathcal{K}$  indicates that pixel  $r$  is labeled as member of class  $k$ . Therefore,  $C(r) = 0$  indicates that pixel  $r$  is unlabeled and its label needs to be estimated; *i.e.*, the region of interest is defined as  $\Omega = \{r \in \mathcal{L} : C(r) = 0\}$ . The segmentation procedure is as follows. Let  $g = \{g_i\}_{i=1}^3$  be the original image (in the RGB space), then we compute at each pixel and for each color layer [with (21)] the local structure tensor  $\Sigma_g(r) = \{\Sigma_1(r), \Sigma_2(r), \Sigma_3(r)\}$ ; where  $\Sigma_i$  is a symmetric semi-positive definite matrix:

$$\Sigma_i(r) = \begin{bmatrix} g_{i11}(r) & g_{i12}(r) \\ g_{i12}(r) & g_{i22}(r) \end{bmatrix} \quad \text{for } i = 1, 2, 3. \quad (24)$$

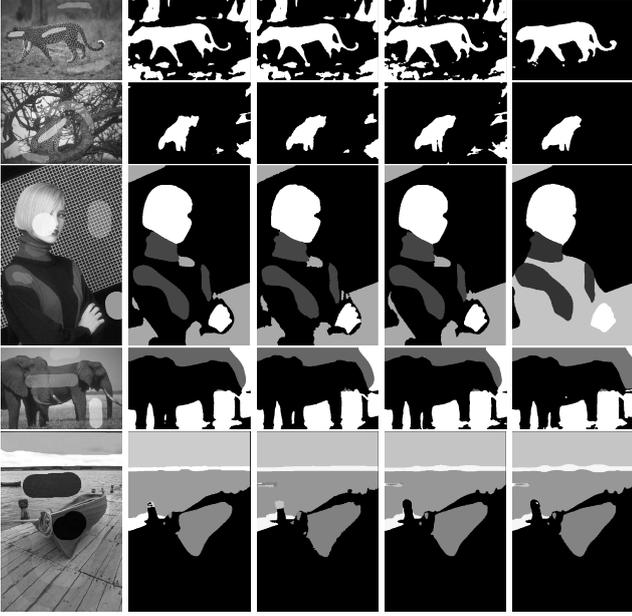
Then, we group the original data and tensor's coefficients in four feature sets that are considered as four independent segmentation sources:

$$g = \{g_i\} \quad (25)$$

$$g_{11} = \{g_{i11}\} \quad (26)$$

$$g_{12} = \{g_{i12}\} \quad (27)$$

$$g_{22} = \{g_{i22}\} \quad (28)$$



**FIGURE 6.** Images segmentation combining multiple clues: original image (first column), GMMF with color clues (second column), QMMF with color clues (third column), MSL with color clues (fourth column) and, MSL with color and texture clues (last column); see text.

Image	Source's confidence			
	$g$	$g_{11}$	$g_{12}$	$g_{22}$
Cheetah	0.2284	<b>0.2714</b>	0.2411	0.2592
Cheetah 2	0.2547	<b>0.2603</b>	0.2276	0.2574
Girl	<b>0.2778</b>	0.2510	0.2283	0.2428
Elefant	<b>0.3280</b>	0.2270	0.2156	0.2294
Port	<b>0.4539</b>	0.1667	0.1998	0.1796

**TABLE 1.** Confidence factors to combine the PS (spatial likelihoods) for the experiments in Figure 6.

for  $i = 1, 2, 3$ . The additional feature sets [(26) to (28)] codify the local texture information. Then, the Likelihood Functions are estimated by histograms with  $64 \times 64 \times 64$  bins and the dynamic range of each feature image is linearly mapped into the interval  $[1, 64]$ . Hence, let  $\mathbf{g} = [g, g_{11}, g_{12}, g_{22}]$  be the vector of images (25)–(28) such that  $\mathbf{g}^{(j)}(r) \in t$ , with  $t = \{t_1, t_2, \dots, t_T\}$  and  $t_i \in \mathbb{R}^3$ , is the vectorial value of the  $r$ -th-pixel in the  $j$ -th image. Then, the regular histogram for the  $k$ -th class is computed with:

$$H_k^{(j)}(t) = \frac{\sum_{s \in \mathcal{L}} \delta(k - C(s)) \delta(\|t - \mathbf{g}^{(j)}(s)\|)}{\sum_{s \in \mathcal{L}} \delta(k - C(s))} \quad (29)$$

where  $\delta$  is the Kronecker's delta function. Hence, the pixel-wise likelihoods are computed with  $v_k^{(j)}(r) = H_k^{(j)}(\mathbf{g}^{(j)}(r))$ . Finally, we compute the final spatial likelihood  $\hat{v}$  with (17).

Now, we explain how to compute the confidence of each Likelihood Function set. The confidence factor of a Likelihood Function set (says the  $j$ -th set) is its capability for predicting the correct pixel class. In our interactive

### Algorithm 2 Source Confidence Factors

- 1: Given the normalized individual likelihood  $\hat{v}^{(j)}(r)$   $\forall r, j$ , then to compute the source confidence scores with

$$\alpha^{(j)} = \sum_k \alpha_k^{(j)},$$

where  $\alpha_k^{(j)}$  is computed with (30).

- 2: Normalize the source confidence:

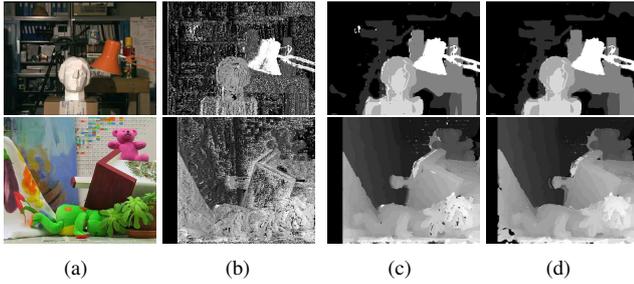
$$\alpha^{(j)} \leftarrow \frac{\alpha^{(j)}}{\sum_l \alpha^{(l)}}.$$

scheme, such a confidence  $\alpha_j$  is large if the likelihoods of the hand-labeled pixels are large for their respective models (and small for the other ones). In particular, we define the confidence of the  $j$ -th source for the  $k$ -th class as follows:

$$\alpha_k^{(j)} = \frac{\sum_r v_k^{(j)}(r) \delta(k - C(r))}{\sum_r \delta(k - C(r))}. \quad (30)$$

If we normalized the individual likelihoods ( $\sum_k v_k^{(j)}(r) = 1, \forall j, r$ ) then  $\alpha_k^{(j)} \approx 1$  represents a high confidence on the source  $j$ -th for predicting the  $k$ -th label. Algorithm 2 details the procedure for computing the source confidence factors.

Figure 6 shows segmented images with the proposed procedure. First column shows the user scribbles on the original image. For comparison purposes, next columns show label maps computed with different algorithms. From second to fourth columns, we show segmentations computed using only the color source ( $g$  in  $rgb$  space). The compared methods are the Gaussian Markov Measure Field (GMMF) model [13], the Quadratic Markov Measure Fields (QMMF) model [15] and the proposed MSL approach, respectively. Last column shows the segmentation computed with the proposed MSL method incorporating color and texture information: the data,  $g$ , and the structure tensor coefficients,  $[g_{11}, g_{12}, g_{22}]$ . The *confidence scores* for the sources were computed with the procedure in Algorithm 2 and are shown in Table 1. This Table shows in bold font the most confident source for each image. Note that if the color source confidence is significantly larger than the texture confidences, then the color based segmentation is qualitatively as good as the one with four sources; those are the cases of the *Elephant* and *Port* images. However, the accuracy of the single source (color) segmentation is reduced as the confidence on such a source is reduced. Indeed, in all the cases the best segmentation was computed with the proposed integration of all the sources. In these experiments and for all the features (sources), the step 4 in Algorithm 1 was implemented with a simple homogeneous membrane (20). In our experiments we set the smoothing parameter  $\lambda = 20$ ; *i.e.*, a wide smoothing filter.



**FIGURE 7.** Layered stereo, columns: a) Left image, b) ML estimator from individual likelihood (no spatial samples), c) MSL estimator using  $\hat{v}$  and d) MSL estimator using color, texture and disparity clues [column (c)].

## 4.2. Layered stereo

From the previous application, we can note that the GHW filter based algorithms are prone to produce smooth class-edge and slightly shifted from the image edges (large gradients). So, as for remaining applications we use SAW based filters.

In order to demonstrate the flexibility of our model, we develop a procedure for computing layered disparities. In this case, SAW filters allow us to prevent disparity over-smoothing because motion edges are highly likely to be allocated at image edges. This strategy has widely been used in other approaches for this task [8, 9, 15]. In this work, we use constant layered motion models, then the disparity models are assumed to be integer-valued and constant:  $m = \{m_k\}_{k=1}^K$ . Then, we define the residual  $d_k(r) \stackrel{def}{=} g_L(r) - g_R(r + m_k)$ ; where  $g_L$  and  $g_R$  are a stereo pair. Now, if we assume Gaussian residuals with mean equals zero, then the likelihoods are computed with

$$v_k(r) = \phi_\sigma(\|d_k(r)\|) \quad (31)$$

where  $\phi$  is the Gaussian kernel defined as:

$$\phi_\sigma(z) \stackrel{def}{=} \exp[-z^2/(2\sigma^2)] \quad (32)$$

and  $\sigma^2$  is the homogeneous variance of the residuals for all the classes. As we said, we compute the confidence between a pair of pixels by taking into account: the spatial, color and likelihood distances. Hence, the  $(r, s)$ -pixels confidence is computed with

$$\omega_{rs} = \phi_{\gamma_1}(\|r - s\|) \phi_{\gamma_2}(\|g(r) - g(s)\|) \phi_{\gamma_3}(v(r)^T v(s)), \quad (33)$$

where the parameters  $\{\gamma_1, \gamma_2, \gamma_3\}$  control the feature contributions. This produces a kind of trilateral filter that it is applied to the log-likelihoods (with uncoupled layers):

$$\hat{v}_k(r) \propto \prod_{s \in \mathcal{N}_r} [v_k(s)]^{\omega_{rs}} = \exp\left(\sum_{s \in \mathcal{N}_r} \omega_{rs} \log v_k(s)\right). \quad (34)$$

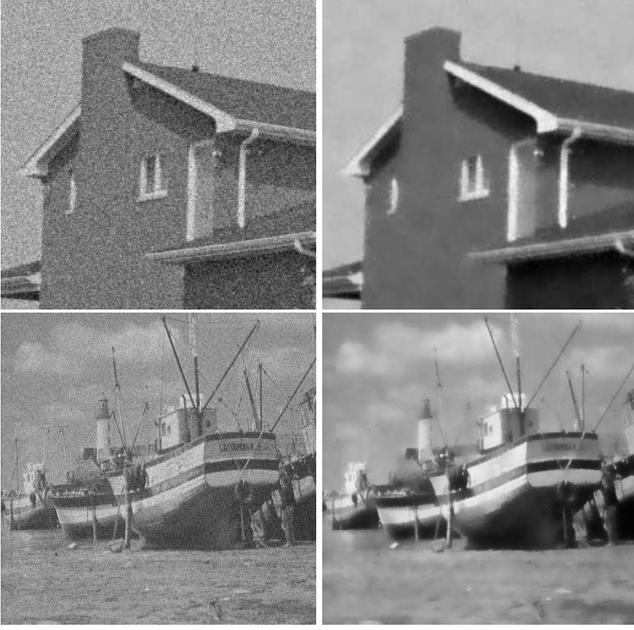
After filtering with (34) and in order to propagate the solution to homogeneous regions and reduce isolated pixels, we apply a soft homogeneous membrane (20) with  $\lambda = 0.3$ .

When a wide spatial neighborhood is used, the combination of color and likelihood confidence factors promote edge preservation. The effectiveness of our approach is demonstrated by our experimental results in Figure 7. First column shows the first and second frames of the sequences *Tsukuba* and *Teddy*, respectively. The data were taken from the Middlebury stereo datasets [31]. Column (b) shows the ML estimator from non spatially filtered likelihoods. Note that the 3D structure of the scene is highly corrupted by impulsive noise. For the *Tsukuba* pair (first 2 frames in the sequence) the set of disparity models were initialized as  $\{m = 4, 5, \dots, 20\}$ . For a second iteration, the models with less than 2% of support were removed. The *Teddy* sequence is computed using the disparity set  $m = \{4, 5, \dots, 45\}$ . Panel (c) shows the results after applying three iterations of the spatial filter (34) with the parameters  $\gamma = \{20.0, 0.1, 0.3\}$  and with neighborhood  $\mathcal{N}_r = \{s : \|r - s\|_\infty \leq 20\}$ .

*Stereo refinement using color and texture clues:* Interactive model initialization (by means of user's scribbles) is a useful technique for solving highly ill-posed problems as, for instance, the segmentation of an image in foreground/background or the general image decomposition. The generative models of such problems have as characteristic of being task depending; *i.e.*, the user may segment a same image in different forms depending of his/her intention. Here we combine the procedures of automatic stereo analysis and interactive segmentation for improving the disparity computation. The resulted method is a fully automatic algorithm (non-interactive). First, we compute the disparity for a stereo pair with the previously presented algorithm. Then, the label map computed from the MSL estimator is used as class scribbles in the procedure developed in Subsection 4.1. Since the disparity map labels all the image pixels (*i.e.*, there are not pixels with  $C = 0$ ) we relax the filter (20) such that all the pixels are updated. Moreover, given that we just want a slight label refinement, the smoothing parameter is set  $\lambda = 0.1$ . Finally, we combine the color, texture (tensor based) and disparity spatial likelihoods with (17) using the fixed confidence factors  $\alpha \propto [0.2, 0.2, 0.2, 0.2, 0.02]$ , where 0.02 is the confidence of the disparity based hard segmentation (MSL) and the remainder weights are for color and texture clues. The results of the post-process are shown in Figure 7, column (d).

## 4.3. Denoising

In this subsection, we present an image denoising method. Our propose is to illustrate other potential application of the our MSL method. Our implementation is based on SAW filters in order to preserve the edges location and small image's structures. Although, the experiments are achieved in grayscale images, our method can straightforward be applied to color images. The gray level (color) models are distributed in the dynamic range of the image according to  $\theta_k = k \times \text{step}$ . In our experiment, we set  $\text{step} = 25$  thus  $k = 0, 1, \dots, 10$ . Then, the individual likelihood were



**FIGURE 8.** Denoising application. Left column: corrupted images with Gaussian noise. Right column: filtered images.

computed with

$$v_k(r) = \phi_{step}(\|g(r) - \theta_k\|), \quad (35)$$

where  $\phi$  is the Gaussian kernel defined in (32). The log-likelihoods are filtered with a combination of a trilateral filter (TF) [Eqs. (33) and (34)] and a membrane filter (MF) [Eq. (20)]. As in the case of stereo analysis, this linear wide-support filter propagates results from high frequency regions to homogeneous areas and controls isolated pixels. The experiments of the denoising technique are shown in Figure 8. The test images were taken from the Portilla's web page [32]. We corrupted the original images with zero-mean Gaussian noise with standard deviation equal 25% of their dynamic range. The reconstruction (filtered images) in the second column are computed with:  $f(r) = \theta^T \hat{v}(r)$ ; where  $\hat{v}(r)$  is computed with three iterations of the TF-MF. We use  $\{\gamma_1 = 1, \gamma_2 = 0.2, \gamma_3 = 0.2\}$  for TF and  $\lambda = 0.5$  for the MF. The parameters are fractions of the image dynamic range.

#### 4.4. Diffusion Multi-Tensor Filtering

The goal of this application is to reconstruct multi-tensor fields that corresponds to trajectories of axonal fibers in white matter of human brain [33]. There are two important issues to consider in this task. In the first, we want to preserve fiber orientation edges and, in the second, we need to allow smooth fiber trajectories. This means that we need to design a filter that smoothes among models (layers) with similar orientations.

One of the most challenging goals in neurosciences is to estimate connectivity patterns for *in vivo* brains. Such connectivity patterns help in the study several brain

diseases and brain development [33, 34]. Diffusion Multi-Tensor Magnetic Resonance Imaging (MTD-MRI) is a neuroimaging technique useful for computing a local (voxel per voxel) estimation of the nerve bundles orientation [34]. That technique assumes a fixed set of Diffusion Basis Functions (DBF). Then, in our notation, the observed signal  $S$  (in the modality Diffusion-Weighted MRI) follows the model:

$$S(\mathbf{q}_n, \tau) = \sum_{k=1}^K v_k \psi_{kn}(\mathbf{q}_n, \tau) + \varepsilon_n; \quad (36)$$

where  $\varepsilon_n$  is a residual, the  $\psi_k$  is the  $k$ -th DBF that corresponds with a signal due to a single fiber oriented with the unitary vector  $\mathbf{u}_k$  (the vectors are uniformly distributed in the 3D orientation space, see Ref. [33]). Then, we fix the DBF by fixing the diffusion base tensors:

$$\bar{\mathbf{T}}_k = (1 - \beta)\mathbf{u}_k\mathbf{u}_k^T + \beta\mathbf{I} \quad (37)$$

where  $\mathbf{I}$  denotes the identity matrix and  $\beta$  is a small scalar, we use  $\beta = 0.1$ . Thus,

$$\psi_{kn}(\mathbf{q}_n, \tau) = S_0 \exp(\tau \mathbf{q}_n^T \bar{\mathbf{T}}_k \mathbf{q}_n).$$

where  $\tau$  (scalar) and  $\mathbf{q}_n$  (the direction of a magnetic gradient) are parameter determined by the acquisition protocol [34]. Hence  $v_j$  denotes the mixture contribution of each DBF to the signal and it can be understood as the membership or the likelihood that at the  $r$  voxel is present a fiber with an orientation parallel to the unitary vector  $\mathbf{u}_k$ . This  $v$  likelihood field is computed using

$$\begin{aligned} v^* &= \underset{v}{\operatorname{argmin}} \|S - \Psi v\|^2 \\ &\text{s.t. } v_k(r) \geq 0; \end{aligned} \quad (38)$$

see [34] for more details.

Here we propose a procedure to spatially filter the likelihood field  $v$ . Our filter procedure needs to take into account that the local likelihood vector can be multi-modal because axonal fibers can cross, merge or split. The filtered likelihood field is computed with  $\hat{v}_k(r) = \exp(d_k^*(r))$ , where  $d^*$  is the minimizer of the membrane energy:

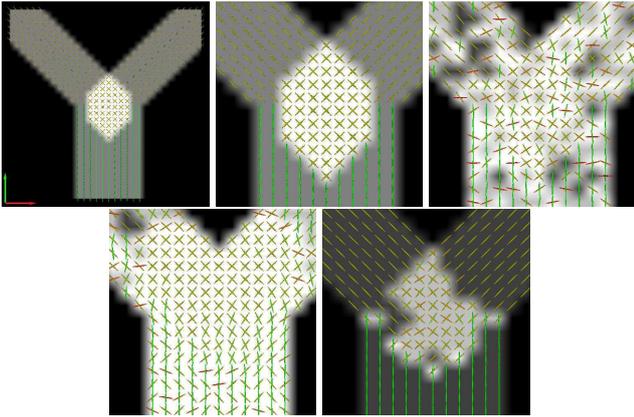
$$\begin{aligned} U(d) &= \sum_{r:C(r)=0} \left\{ \|d(r) - \log \hat{v}^*(r)\|^2 \right. \\ &\quad \left. + \frac{\lambda}{2} \sum_{s \in \mathcal{N}_r} \sum_{k,l} w_{kl} [d_k(r) - d_l(s)]^2 \right\}. \end{aligned} \quad (39)$$

where  $\hat{v}^*$  is the voxelwise normalized version of the likelihood field  $v$ . It is important to note that the weights

$$w_{kl} = \exp[\kappa(\mathbf{u}_k^T \mathbf{u}_l)^2 - \kappa]$$

control the amount interaction between the  $k$  and  $l$  layers (smoothness), where  $\kappa$  is a positive parameter. Then  $W = [w]_{k=1,2,\dots,K;l=1,2,\dots,K}$  is the confusion matrix between the models (fiber orientations).

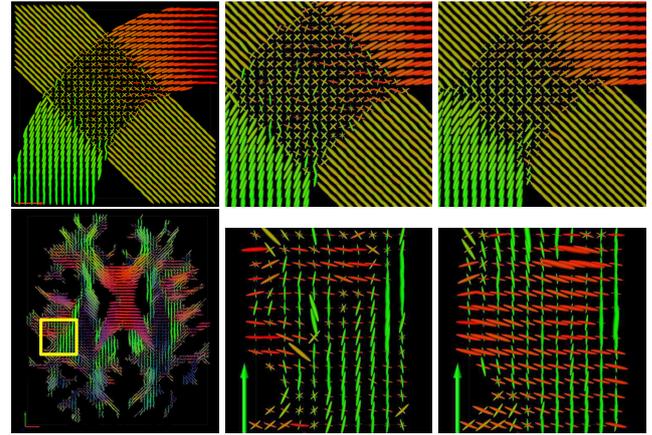
The method performance is demonstrated by experiments with synthetic and real multi-tensor fields. Figure 9 shows



**FIGURE 9.** Diffusion Multi-Tensor restoration. First row: Ground truth, region of interest in the ground truth, noisy data (SNR=9). Second Row: GMMF restoration, restoration with the proposed method.

results computed with our proposal and a multi-tensor restoration based on the GMMF model [13]. The panels show the diffusion tensors on the anisotropy map of the  $S$  signal: white represents low anisotropy and light gray means high anisotropy, see details in Refs. [33, 34]. First row shows the ground truth, the region of interest in the ground truth and the noisy multi-tensor synthetically generated using the method reported in [33] with SNR=9. Second row shows the multi-tensor reconstructions using the GMMF based algorithm (just directly regularizing  $\hat{v}^*$  instead of  $\log v^*$ ) and the proposed method. Note in the ground truth that the lower anisotropy is allocated at the fiber crossing, meanwhile a single fiber produces higher anisotropy. When the data are noise corrupted then the anisotropy coefficient is low at almost all places and false fiber crossings appear (right panel in the first row). The solution using GMMF, basically, diffuses the low anisotropy in the entire image; *i.e.*, it does not reduce the number of crosses (left image in the second row). On the other hand, our proposal increases the anisotropy in almost the whole image. Thus, our proposal correctly estimates the fiber crossing and the sites with a single fiber.

Next we show experimental results of our method in both synthetic and real data. The first row in Figure 10 shows the results on a synthetic multi-tensor data that simulate a fiber crossing. Note that, the restored multi-tensor fields (right panel) show the correct orientations and smooth transitions between orientation. This effect is achieved with the inter-model interaction proposed. The second row shows results with multi-tensors from a real human brain. This figure depicts a region of interest that corresponds to an axon fiber crossing within the *superior longitudinal fasciculus*. Note how our proposal improves the crossing fiber orientations, this is particularly important for the brain axon tractography task.



**FIGURE 10.** Comparison of Diffusion Multi-Tensor restorations. First row: synthetic crossing fibers (multi-tensor field), region of interest and restored region of interest. Second row (from left to right): Real multi-tensor diffusion MRI data, region of interest with fiber crossing and restored (filtered) multi-tensor field in the region of interest.

## 5. CONCLUSIONS

We presented a novel framework for image Probabilistic Segmentation (PS). The presented technique is a probabilistic segmentation strategy: instead of computing the pixel label, we compute the uncertainty (probability) associated to each particular label. We start our development by noting that the spatial sampling is an alternative to the lack of multiple pixels' observations. Differently from multiple observations, the pixel neighborhood is a mixed sample and the estimated mixture coefficients can be used as the probability for the labels. We noted that the neighborhood selection is an important issue to obtain a good segmentation. Moreover, the mixture coefficients must be as informative as possible in order to reduce the risk of choosing a wrong label.

We presented a set of PS algorithms that constructs on well-known spatial filtering techniques. The novelty is the spatial filtering of log-likelihood. Our strategy allows one to combine multiple sources (probabilistic segmentations) in a natural way and is general enough to be applied in the development of algorithms for different computer vision applications. In particular, we have explored interactive segmentation, stereo disparity computation, image denoising and reconstruction of brain water diffusion multi-tensor fields. The experiment results demonstrate that our direct implementations are competitive with *ad hoc* sophisticated algorithms of the state of the art.

A preliminary version of this work appeared in the Proc. of the ISCIS 2010 [35].

## FOUNDING

This research was partially supported by the CONACYT, Mexico, [61367-Y,131369 to M.R. and PhD. Scholarship to O.D.] and the CONCYTEG, Guanajuato Mexico, [09-16-K662-075 to A. R-M.]

## REFERENCES

- [1] Olsson, C., Eriksson, A. P., and Kahl, F. (2008) Improved spectral relaxation methods for binary quadratic optimization problems. *Computer Vision and Image Understanding*, **112**, 30–38.
- [2] Weiss, Y. (1999) Segmentation using eigenvectors: A unifying view. *Proc. 7th Int. Conf. Computer Vision (ICCV'99)*, vol. 2, Kerkyra, Greece, 20-25 September, pp. 975–982. IEEE CS Press, Los Alamitos, CA.
- [3] Shi, J. and Malik, J. (2000) Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, **22**, 888–905.
- [4] Levin, A., Rav-Acha, A., and Lischinski, D. (2008) Spectral matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, **30**, 1–14.
- [5] Li, S. Z. (2001) *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, Tokyo.
- [6] Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, **6**, 721–741.
- [7] Alarcón, T. E. and Marroquín, J. L. (2009) Linguistic color image segmentation using a hierarchical Bayesian approach. *Color Research & Application*, **34**, 299–309.
- [8] Kolmogorov, V., Criminisi, A., Blake, A., Cross, G., and Rother, C. (2006) Probabilistic fusion of stereo with color and contrast for bi-layer segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**, 1480–1492.
- [9] Blake, A., Rother, C., Brown, M., Perez, P., and Torr, P. (2004) Interactive image segmentation using an adaptive GMMRF model. *Proc. European Conference in Computer Vision (ECCV'04)*, Prague, Czech Republic, 11–14 May, pp. 428–441. Springer-Verlag, Berlin.
- [10] Boykov, Y., Veksler, O., and Zabih, R. (2001) Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, **23**, 1222–1239.
- [11] Boykov, Y. and Jolly, M.-P. (2001) Interactive graph cut for optimal boundary & region segmentation of objects in N-D images. *Proc. Eighth Int. Conf. Computer Vision (ICCV'01)*, vol. 1, Vancouver, BC, 7-14 July, pp. 105–112. IEEE CS Press, Los Alamitos, CA.
- [12] Rother, C., Kolmogorov, V., and Blake, A. (2004) Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, **23**, 309–314.
- [13] Marroquin, J. L., Velazco, F., Rivera, M., and Nakamura, M. (2001) Gauss-Markov measure field models for low-level vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**, 337–348.
- [14] Grady, L. (2006) Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**, 1768–1783.
- [15] Rivera, M., Ocegueda, O., and Marroquin, J. L. (2007) Entropy-controlled quadratic Markov measure field models for efficient image segmentation. *IEEE Trans. Image Processing*, **8**, 3047–3057.
- [16] Rivera, M., Dalmau, O., and Tago, J. (2008) Image segmentation by convex quadratic programming. *Proc. Int. Conf. on Pattern Recognition (ICPR'08)*, Tampa, FL, 8–11 December, pp.1–5. IEEE CS Press, Los Alamitos, CA.
- [17] Sfikas, G., Nikou, C., Galatsanos, N., and Heinrich, C. (2010) Spatially varying mixtures incorporating line processes for image segmentation. *J. Math. Imaging Vis.*, **36**, 91–110.
- [18] Dempster, A., Laird, N., and Rubin, D. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, **39**, 1–38.
- [19] Greig, D., Porteous, B., and Scheult, A. (1989) Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, **51**, 271–279.
- [20] Kolmogorov, V. and Zabih, R. (2004) What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**, 147–159.
- [21] Kohli, P. and Torr, P. H. S. (2008) Measuring uncertainty in graph cut solutions. *Computer Vision and Image Understanding*, **112**, 30–38.
- [22] Komodakis, N., Tziritas, G., and Paragios, N. (2008) Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal–dual strategies. *Computer Vision and Image Understanding*, **112**, 14–29.
- [23] Marroquin, J. L., Arce, E., and Botello, S. (2003) Hidden Markov measure field models for image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, **25**, 1380–1387.
- [24] Kohli, P. and Torr, P. H. S. (2007) Dynamic graph cuts for efficient inference in Markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**, 2079–2088.
- [25] Gonzalez, R. C. and Woods R. E. (2008) *Digital Image Processing 3rd Ed.* Pearson Prentice Hall, Upper Saddle River, NJ.
- [26] Perona, P. and Malik, J. (1990) Scale-space and edge-detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, **12**, 629–639.
- [27] Terzopoulos, D. (1986) Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**, 413–424.
- [28] Weickert, J. (1999) Coherence-enhancing diffusion filtering. *Int'l J. Computer Vision*, **31**, 111–127.
- [29] Tomasi, C. and Manduchi, R. (1998) Bilateral filtering for gray and color images. *Proc. Sixth International Conference on Computer Vision (ICCV'98)*, Bombay, India, 4-7 January, pp. 839–846. Narosa Publishing House, New Delhi, India.
- [30] Barash, D. (2002) A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**, 844–847.
- [31] Scharstein, D. and Szeliski, R. (2003) High-accuracy stereo depth maps using structured light. *Proc. Int. Conf. on Comp. Vis. Patt. Recog. (CVPR'03)*, Madison, WI, 16–22 June, pp. 195–202. IEEE CS Press, Los Alamitos, CA.
- [32] Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E. (2003) Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Proc.*, **12**, 1338–1351.
- [33] Ramirez-Manzanares, A., Rivera, M., Vemuri, B. C., Carney, P., and Mareci, T. (2009) Diffusion basis functions decomposition for estimating white matter intravoxel fiber geometry. *IEEE Trans. Med. Imaging*, **26**, 1091 – 1102.
- [34] Ramirez-Manzanares, A., Rivera, M., and Gee, J. (2009) Denoising intra-voxel axon fiber orientations by means of ECQMMF method. *Proc. Mexican Int. Conf. Artificial Intell. (MICAI'09)*, Guanajuato, Mexico, 9–13 November, pp. 303–312. Springer, Berlin.
- [35] Rivera, M., Dalmau, O. and Washington M. (2010) Spatial Sampling for Image Segmentation. *Proc. 25th Int. Symp. Computer and Information Sciences (ISCIS'10)*, London, UK, 22–24 September, pp. 309-314, Springer, Netherlands.