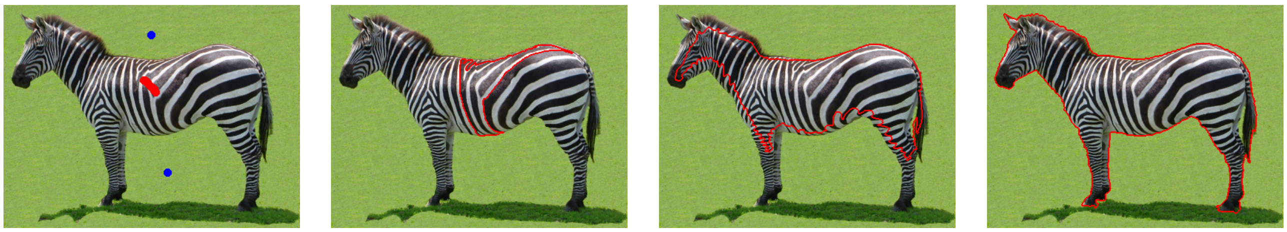
Multivalued label diffusion for semi-supervised segmentation

Pierre Buyssens, Olivier Lézoray

MULTIVALUED LABEL DIFFUSION FOR SEMI-SUPERVISED SEGMENTATION

Pierre Buyssens, Olivier Lézoray

Normandie Université, UNICAEN, ENSICAEN, GREYC UMR CNRS 6072, Caen, France



Initial seeds | Result with [1] | Result with [2] | This paper

ABSTRACT

Diffusion methods have proven their efficiency for tasks such as semi-supervised segmentation. The introduction of patches as a part of their speed function allows to deal with textured images. However, the computational burden with such variants stays too important for low-level tasks. In this paper, we propose a multivalued color-based potential function that partly alleviates this flaw. It allows to efficiently perform semi-supervised segmentation of natural and textured images.

***Index Terms*—** Diffusion, Gaussian Mixture Models, Semi-supervised segmentation, Textured images

1. INTRODUCTION

Diffusion based approaches have been widely used for image segmentation purposes [3, 4, 5, 1, 7] (to cite a few). Allowing a user to label parts of an image leads to a semi-supervised segmentation problem, for which most of label diffusion techniques can be used. Apart from their different formulations, their results strongly differ with the amount of manual interactions needed to get a good segmentation result.

The *zebra* teaser images show that sparsely distributed seeds (first image) are clearly insufficient even with a state-of-the-art method such as [1] (second image). Indeed [1] requires far more seeds to correctly segment the zebra, and that is the case for most of diffusion-based segmentation approaches: the quality of the result heavily depends on the nature of the image and on the amount of user's interactions.

In this paper we propose a multivalued diffusion framework whose goal is to reduce the amount of user interactions to get a satisfactory result while keeping its complexity low.

The proposed method[1] relies on the resolution of an Eikonal equation that computes for each pixel p of an image I the distance $U(x)$ to its closest seed pixel. It then defines a partition of the image leading to the segmentation result.

The proposed method extends our previous work [2] to handle a multivalued diffusion behavior. The considered equation is:

$$\begin{cases} |\nabla U(x)| \;=\; P(u, L_t) & \forall p \in I \\ U(x) \;=\; \phi(x) & \forall x \in L_0 \end{cases} \tag{1}$$

where P is the positive potential function, ϕ an initialization function, L_t the set of labeled pixels at the t^{th} iteration, and L_0 the set of initial labeled pixels.

Contrary to classical gradient-based potential function, where P is fully known at the beginning of the diffusion, we propose the use of a dynamic potential computed according to statistics of the regions while they grow.

To efficiently deal with multivalued objects, a region is no more modeled with its mean as in [2], but with Gaussian Mixture Models (GMM) such that it may be composed of very different features. On the *zebra* teaser image for instance, the initial seed depicted in red is roughly composed of white and black components. The region (front) can then grow freely along these two colors and this results in a correct segmentation (right image). Note that such GMMs have previously been used within the Graph cut optimization framework [8]. However our proposed approach has a much lower complexity.

2. PROPOSED MULTIVALUED DIFFUSION

2.1. Notations

In the following, a pixel p consists in a couple of coordinates (x_p, y_p) and an associated dimensional feature vector $\mathbf{X}_p \in \mathbb{R}^d$ ($d = 3$ for color images). A region R_i is a

[1]code available at `sites.google.com/site/pierrebuyssens/code/multivalued-diffusion`

set of pixels enclosed within a front $\Gamma_i = \{q \in R_i | \exists p \notin R_i \text{ and } p \text{ neighbor of } q\}$. The Gaussian Mixture Model associated to R_i is composed of K Gaussians and is noted $\Pi_i = \sum_{k=1}^{K} \pi_k G_k^i$ where G_k^i is the k^{th} Gaussian component of the mixture and π_k its mixture coefficient. We assume that the mixture coefficients for a given GMM sum to 1 (i.e., $\sum_{k=1}^{K} \pi_k = 1$). In the following, we detail the algorithm considering only one particular GMM. So, for clarity purposes we drop the index i. A given Gaussian G_k consists in a mean vector $\boldsymbol{\mu}_k$, a full $d \times d$ covariance matrix $\boldsymbol{\Sigma}_k$, and the number of samples n_k associated to it.

2.2. Gaussian Mixture-based potential function

In this paper, we propose a dynamic potential function that depends on statistics of the region begin formed. Assuming that a Gaussian Mixture Model Π composed of K Gaussians is associated to a given region R

$$\Pi = \sum_{k=1}^{K} \pi_k G_k \tag{2}$$

with

$$G_k(\mathbf{X}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_\mathbf{k}|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{X}-\boldsymbol{\mu}_k)\right)}. \tag{3}$$

Then the potential $P(p|R)$ of a pixel $p \in \Gamma$ is computed as

$$P(p|R) = \frac{1}{\gamma(p|R)} \tag{4}$$

where

$$\gamma(p|R) = \text{argmax}_k(\pi_k G_k(\mathbf{X}_p)). \tag{5}$$

Here, the γ function can be seen as a speed function of the front Γ for a given pixel p. Equation (5) states that only the Gaussian that best fits $\boldsymbol{X}_p$ is associated to p (hard assignment[2]). At this step, one can note that a Gaussian component G_k that is under represented within the mixture (i.e., $\pi_k \ll 1$) would produce low speed values $\pi_k G_k(\cdot)$. The front diffusion through such a Gaussian is then not fostered.

2.3. Algorithm detail

Initialization : Given a set of initial labeled pixels, the algorithm starts by modeling each set of pixels that share the same label by a Gaussian Mixture Model. Each GMM is initially composed of K components ($K = 10$ in all our experiments). For a given GMM, each mean μ_k is estimated via k-means and a sole Gaussian is assigned to each initial labeled pixel. Covariance matrices Σ_k are then computed with respect to this assignment. Finally, the mixture coefficients are simply computed as the probability for a pixel to belong to a given Gaussian : $\pi_k = n_k / \sum_{t=1}^{K} n_t$.

[2] Soft assignment of probabilities would be preferable to be used. However the practical benefits of such assignments turn out to be neglectible regarding the additional computational cost.

Note that irrelevant Gaussians (those that contain no pixels, $n = 0$) are automatically discarded. An initial GMM can then be composed of less than K components.

Diffusion : The diffusion process is carried out via the Fast Marching algorithm due to its efficiency [9]. The potential of each pixel $p \in \Gamma_i, i \in \{1, \dots, N\}$ is computed (Eq. (4)). Within the diffusion process, each time a pixel p is labeled, the adjoining Gaussian G (the one that has maximized $\pi_k G_k$, see Eq. (5)) is updated with $\mathbf{X}_p$. To avoid recomputing from scratch the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$ of G, which can be *very* time consuming, we perform an online update of G. This non-trivial update procedure of a given Gaussian G with a feature vector $\mathbf{X}$ is detailed in the algorithm 1. Note that the mixture coefficients of all the Gaussians of the GMM are updated too, since we have $\sum_{k=1}^{K} \pi_k = 1$. The algorithm terminates when all the pixels have been labeled.

Algorithm 1 Online update of a Gaussian G by a vector $\mathbf{X}$

Input: $\mathbf{X}, G$
Output: updated G
$\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma} \times (n-1)$
$n \leftarrow n + 1$
$\boldsymbol{\Delta} \leftarrow \mathbf{X} - \boldsymbol{\mu}$
{Update mean $\boldsymbol{\mu}$}
$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \boldsymbol{\Delta}/n$
{Update covariance matrix $\boldsymbol{\Sigma}$}
$\boldsymbol{\Lambda} \leftarrow \mathbf{X} - \boldsymbol{\mu}$
$\boldsymbol{\Sigma}_{\mathbf{i},\mathbf{i}} \leftarrow \boldsymbol{\Sigma}_{\mathbf{i},\mathbf{i}} + \boldsymbol{\Delta}_\mathbf{i} \times \boldsymbol{\Lambda}_\mathbf{i}$
$\boldsymbol{\Sigma}_{\mathbf{i},\mathbf{j}} \leftarrow \boldsymbol{\Sigma}_{\mathbf{i},\mathbf{j}} + \boldsymbol{\Lambda}_\mathbf{i} \times \boldsymbol{\Lambda}_\mathbf{j}$ if $i \neq j$
$\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma}/(n-1)$
{Update mixture coefficients $\pi_k, \forall\, G_k$}
$\pi_k \leftarrow n_k / \sum_{t=1}^{K} n_t$

2.4. Complexity analysis

Complexity of the diffusion relies essentially on the complexity of the Fast Marching algorithm and on the number of Gaussians. With an appropriate heap for sorting the pixels according to their potential, the complexity of the Fast Marching algorithm depends on the number of pixels D and is roughly $\mathcal{O}(D\log(D))$. Also, computing the potential of a pixel p involves K estimations of $\pi_k G_k(\mathbf{X}_p)$ (Eq. (5)). The overall complexity of the proposed algorithm is then roughly $\mathcal{O}(KD\log(D))$.

Despite this theoretical complexity, the proposed algorithm may be very fast in practice, since the computation of $\gamma(p|R)$ (Eq. (5)) can easily be parallelized. Moreover, using different data structures (and additional storage), some $\mathcal{O}(D)$ implementations of the Fast Marching algorithm have been proposed [10, 11]. With a smart implementation, the complexity of the proposed algorithm can then be reduced to $\mathcal{O}(D)$. This linear complexity is a strong advantage towards the use of our approach.

3. RESULTS

3.1. Multi label semi-supervised segmentation

In this section, we compare segmentation results of our approach with state-of-the-art methods: a gradient-based diffusion framework [12], the Eikonal Region Growing Clustering algorithm[3] [2], and the Power Watershed algorithm[4] [1]. While the gradient-based diffusion framework [12] solves an Eikonal equation with a gradient-based potential function, the power watershed [1] uses a variant of the watershed algorithm on graph. Finally, our previous work [2] solves an Eikonal equation with a potential function that depends solely on the mean color of the regions.

The comparative results on the *church* image (left column of Fig. 1) show many bad labeling for state-of-the-art methods, especially on the front of the building or on the roof. On the *landscape* image (middle column), similar failures can be observed, especially on stones and grass. Finally, the segmentation results on the *Oscar* image (right column) exhibits bad labeling with methods of [12] and [2]. The power watershed performs well on this image.

Despite sparsely and coarsely distributed initial labels, our GMM based diffusion approach provides equal or better quality visual results than state-of-the-art methods.

3.2. Adding texture informations

The proposed formulation (see Sec. 2) is not restricted to color feature vectors. In this section, we provide additional segmentation results on texture images by adding simple texture information to the model. Introduced in [13], *structure tensors* are a natural extension of gradient motion for multi-valued images. They efficiently encode both the local image color variation and their directions. Traditionnally, for two-dimensional RGB images, structure tensors reduce to 2×2 matrices defined as follows:

$$\begin{aligned}\mathbf{S} &= \sum_{c \in \{R,G,B\}} \overrightarrow{\nabla I^c} . \overrightarrow{\nabla I^c}^T \\ &= \lambda_1 . \mathbf{u} . \mathbf{u}^T + \lambda_2 . \mathbf{v} . \mathbf{v}^T \quad \text{with} \quad \lambda_1 > \lambda_2\end{aligned}$$

with $\lambda_{\{1,2\}}$ the eigenvalues of $\mathbf{S}$ and $\mathbf{u}$, $\mathbf{v}$ the eigenvectors associated to λ_1 and λ_2 respectively. The eigenvector associated to the largest eigenvalue is oriented along the major image change direction while the one associated to the smaller eigenvalue is oriented along its orthogonal.

In the following, we use the largest eigenvalue and its associated eigenvector as additional components in our model. The feature vector associated to each pixel consists then in a 6-dimensional vector encoding both color and local texture informations.

[3] sites.google.com/site/pierrebuyssens/code/ergc

[4] powerwatershed.sourceforge.net

Except for these augmented inputs, the rest of the algorithm is kept unchanged. Figure 2 provides comparisons of state-of-the-art algorithms with ours on synthetic textured images. Despite sparsely and coarsely distributed initial seeds, the proposed approach efficiently discriminates different orientations of the same texture (first column), similar patterns with different luminance (second column), and more complex texture patterns (third column). All other methods are not suited for such complex images.

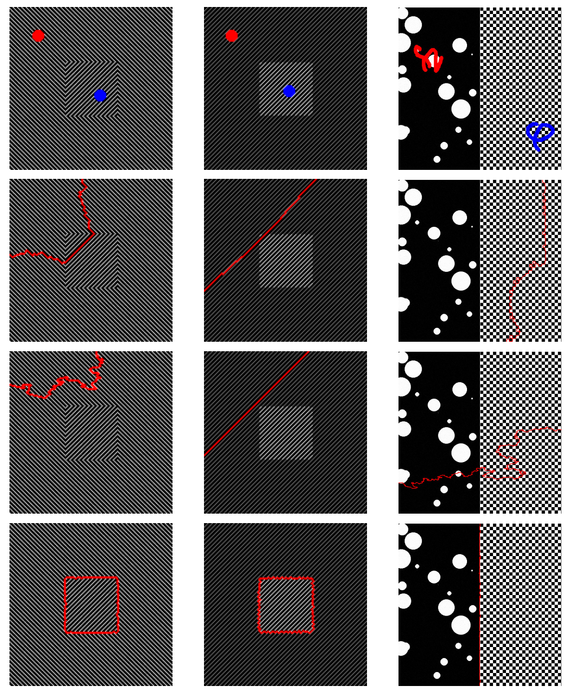


Fig. 2: Semi-supervised segmentation of textured images. From top to bottom: initial seeds, segmentation results with [2], [1], and our approach.

4. CONCLUSION AND FUTURE WORK

This paper presents a multivariate diffusion framework for semi-supervised segmentation. Based on a diffusion from on a set of initial seeds, the proposed approach allows to segment complex objects that are modeled with Gaussian Mixture Models. We also demonstrate the flexibility of our approach by adding simple structure features to the color ones to efficiently deals with texture images. Further work will investigate the extension to image matting such as [7], and its adaptation on graphs for data clustering such as the recent methods [14, 15, 16].

Fig. 1: Semi-supervised segmentation comparisons on the *church* image (left), on the *landscape* image (middle), and on the *Oscar* image (right). From top to bottom: initial labels, segmentation results with a gradient-based potential function [12], segmentation results with a color and region-based potential function [2], segmentation results with power watersheds [1], and segmentation results with the proposed Gaussian Mixture Model potential function.

5. REFERENCES

[1] Camille Couprie, Leo Grady, Laurent Najman, and Hugues Talbot, "Power watershed: A unifying graph-based optimization framework," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 7, pp. 1384–1399, 2011.

[2] Pierre Buyssens, Isabelle Gardin, Su Ruan, and Abderrahim Elmoataz, "Eikonal-based region growing for efficient clustering," *Image and Vision Computing*, 2014.

[3] Mikaël Rousson, Thomas Brox, and Rachid Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," in *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*. IEEE, 2003, vol. 2, pp. II–699.

[4] Leo Grady, "Random walks for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 11, pp. 1768–1783, 2006.

[5] Christopher Alvino, Gozde Unal, Greg Slabaugh, Bertrand Peny, and Tong Fang, "Efficient segmentation based on eikonal and diffusion equations," *International Journal of Computer Mathematics*, vol. 84, no. 9, pp. 1309–1324, 2007.

[6] V.-T. Ta, O. Lezoray, A. Elmoataz, and S. Schupp, "Graph-based tools for microscopic cellular image segmentation," *Pattern Recognition Special Issue on Digital Image Processing and Pattern Recognition Techniques for the Detection of Cancer*, vol. 42, no. 6, pp. 1113–1125, 2009.

[7] Xue Bai and Guillermo Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[8] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[9] James A Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

[10] Seongjai Kim, "An $\mathcal{O}(n)$ level set method for eikonal equations," *SIAM journal on scientific computing*, vol. 22, no. 6, pp. 2178–2193, 2001.

[11] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro, "$\mathcal{O}(n)$ implementation of the fast marching algorithm," *Journal of computational physics*, vol. 212, no. 2, pp. 393–399, 2006.

[12] X. Desquesnes, A. Elmoataz, and O. Lézoray, "Eikonal equation adaptation on weighted graphs: fast geometric diffusion process for local and non-local image and data processing," *Journal of Mathematical Imaging and Vision*, vol. 46, no. 2, pp. 238–257, 2013.

[13] Silvano Di Zenzo, "A note on the gradient of a multi-image," *Computer vision, graphics, and image processing*, vol. 33, no. 1, pp. 116–125, 1986.

[14] E Merkurjev, T Kostic, and A Bertozzi, "An mbo scheme on graphs for segmentation and image processing.," *SIAM J. Imaging Sci.*, vol. 6(4), pp. 1903–1930, 2013.

[15] Xavier Bresson, Thomas Laurent, David Uminsky, and James von Brecht, "Multiclass total variation clustering," in *Advances in Neural Information Processing Systems*, 2013, pp. 1421–1429.

[16] Cristina Garcia-Cardona, Ekaterina Merkurjev, Andrea L Bertozzi, Arjuna Flenner, and Allon G Percus, "Fast multiclass segmentation using diffuse interface methods on graphs," Tech. Rep., DTIC Document, 2013.