15-06
15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

An MRF-based image segmentation with unsupervised model parameter estimation

Yoshihiko Toya
Graduate School of Systems and Information Engineering, University of Tsukuba
1-1-1 Tennodai, Tsukuba 305-8573, Japan
toya@imagelab.cs.tsukuba.ac.jp

Hiroyuki Kudo
Faculty of Engineering, Information and Systems, University of Tsukuba
1-1-1 Tennodai, Tsukuba 305-8573, Japan
kudo@cs.tsukuba.ac.jp

Abstract

This paper deals with image segmentation when the image consists of uniform background (b.g.) and uniform foreground (f.g.) with noise. We formulate this problem into the joint minimization of MRF energy with respect to a label image and density parameters corresponding to f.g. and b.g., and solve it exactly in reasonable computation time. The proposed method efficiently solves the joint minimization by utilizing the novel property that multiple minimizations of MRF enegy, corresponding to different combinations of density parameters for b.g. and f.g., can be solved by a single total-variation minimization. In addition, we also extend the proposed method to the case where label images together with density values corresponding to multiple smoothing (regularization) parameters can be obtained exactly and simultaneously with a much shorter computation time compared with the trivial exhaustive search.

1 Introduction

Image segmentation aims at dividing an image into target regions (foreground, f.g.) and the others (background, b.g.). It is an important step for various image analysis tasks. This problem can be formulated as an optimization problem that defines an energy function and minimizes it. In particular, the optimization method based on MRF model is robust against noise, because it takes the correlation between adjacent pixels into account. The graph cut technique [2] can obtain a global optimum solution of the optimization problem in polynomial time, and has been actively studied in recent years. Alternatively, Tayama and Kudo [1] has proposed a simple continuous optimization framework using convex relaxation and subgradient method.

In the MRF-based image segmentation, a user is required to input three parameters, namely, two mean density values corresponding to f.g. and b.g., and a smoothing (regularization) parameter. In many cases, however, these values are unknown, and we must estimate them using only a given image in an unsupervised way. In this paper, we propose a new image segmentation method, which is able to estimate the parameters without any *a priori* information (in an unsupervised way) under the assumption that the density values corresponding to f.g. and b.g. are uniform.

2 Image segmentation

Image segmentation can be considered as a problem of assigning a binary label $\{0,1\}$ to each pixel of a given image $\vec{y} \in \mathbb{R}^N$ and outputting a label image $\vec{x} \in \{0,1\}^N$.

The image segmentation is often solved as an optimization problem of minimizing an energy function designed based on the MRF model. The energy function of segmentation is defined as follows, when f.g. and b.g. have constant densities.

$$f(\vec{x}) = (1-\beta)\frac{1}{2}\sum_{u \in V} g_u(x_u) + \beta \sum_{(u,v)\in E} h_{uv}(x_u, x_v), \tag{1}$$

where V is a set of pixels, E is a set of pairs of adjacent pixels, and x_u is a label of pixel u. The first term is called the data term, which evaluates similarity between the density of each pixel in the measured image and the mean density corresponding to each region, *i.e.* f.g. and b.g.

$$\begin{aligned} g_u(x_u) &= \begin{cases} |y_u - m_0|^2 & (\text{if } x_u = 0) \\ |y_u - m_1|^2 & (\text{if } x_u = 1) \end{cases} \qquad (2) \\ &= |y_u - m_0|^2(1-x_u) + |y_u - m_1|^2 x_u, \end{aligned}$$

where y_u is the density of pixel u in the measured image, (m_0, m_1) are means of density values corresponding to b.g. and f.g. The squared error between y_u and the respective mean m_0 (or m_1) is used as an energy value. The second term is known as the smoothing (regularization) term, which can be explicitly written as follows.

$$\begin{aligned} h_{uv}(x_u, x_v) &= w_{uv} \times \begin{cases} 0 & (\text{if } x_u = x_v) \\ 1 & (\text{if } x_u \neq x_v) \end{cases} \qquad (3) \\ &= w_{uv}|x_u - x_v| \end{aligned}$$

$$w_{uv} = \frac{1}{\mathrm{dist}(u,v)} \tag{4}$$

The data term focuses on only the density value of each pixel, so that it generates a degraded noisy segmentation result. The aim of smoothing term is to reduce the influence of degradation by using the assumption that adjacent pixels are likely to have the same label. The parameter $\beta \in (0,1)$ in eq. (1) is a hyper-parameter to adjust the strength of each term.

2.1 Unsupervised estimation of density parameters (Scenario 1)

In the above explanation, when solving the energy minimization problem, we assumed that the measured image $\vec{y}$, the smoothing parameter β, and the density parameters (m_0, m_1) corresponding to b.g. and f.g. are given by a user. In many cases, however, (m_0, m_1) are unknown. Therefore, in the conventional segmentation methods, they need to be estimated by using some *a prior* information related to the density values. In [6], Boykov and Jolly used the histogram of measured image, where (m_0, m_1) were estimated by using sets of pixels called "seeds" given by the user. In [7], Zagrouba *et al.* proposed an automatic segmentation method dedicated for flower images using some spatial information. Then they estimated the color distribution and segmented it precisely. On the other hand, Otsu's method [9], which is a classical binary image segmentation method, determines an optimum threshold value from the histogram of measured image and performs the segmentation, where it is not necessary to input (m_0, m_1). However, because Otsu's method does not take the correlation between adjacent pixels into account, the segmented result is significantly affected by the noise in measured image. Therefore, in this work, we propose a segmentation method that can reduce the influence of noise without using *a priori* information such as the seeds and the user's input of (m_0, m_1). To perform the segmentation without (m_0, m_1), we formulate the problem as the following joint minimization of label image $\vec{x}$ and the density parameters (m_0, m_1).

$$\min_{m_0, m_1} \min_{\vec{x}} f(\vec{x}) \tag{5}$$

By solving $\min_{\vec{x}} f(\vec{x})$ for all candidates of (m_0, m_1) values and taking the minimum among all the obtained energy values, we can exactly obtain the solution of eq. (5). However, this exhaustive search method is infeasible, because, when the number of image gray levels G is 256, there exist $\binom{G}{2} = 32640$ combinations of (m_0, m_1). Since the conventional MRF segmentation already involves an iterative computation, the exhaustive search method is not a practical way to perform the unsupervised segmentation.

2.2 Generating segmentations with multiple smoothing parameters (Scenario 2)

In addition, a resonable value of smoothing parameter β is also non-obvious. In many cases, the value of β is determined by trial-and-error, *i.e.* the segmentation is performed with a number of β values and the best value is selected by the user. However, it is inefficient to perform the segmentation many times until getting a satisfactory result. In this work, we also propose a method which generates multiple segmentation results corresponding to a number of different β values simultaneously in an efficient way.

In the left part of this paper, we refer the segmentation for a single smoothing parameter β (given by the user) to as Scenario 1, and the one which produces multiple segmentation results for a number of β values to as Scenario 2.

3 Proposed method

3.1 Total variation and parametric maximum flow

The proposed segmentation method utilizes the Total Variation (TV) minimization as a key tool [3–5]. The TV is normally used as a regularization term to denoise an image contaminated with noise.

$$\mathrm{TV}(\vec{x}) = \sum_{(u,v)\in E} |x_u - x_v| \tag{6}$$

$$f(\vec{x}) = \beta\,\mathrm{TV}(\vec{x}) + \frac{1}{2}\|\vec{x} - \vec{y}\|^2, \tag{7}$$

where eq. (6) is the TV term. By minimizing the energy function of eq. (7), the denoised image $\vec{x}$ can be obtained from the degraded image $\vec{y}$. It is well-known that this problem can be solved efficiently by using the so-called Chambolle's projection algorithm [4, p.316 Algorithm 1].

If $J(\vec{u})$ is the TV term, the following proposition holds.

Proposition 1 ([3, p.292 Proposition 3.3]) *Let $\vec{g} \in \mathbb{R}^N$ and let $\vec{u} \in \mathbb{R}^N$ be the (unique) solution of*

$$\min_{\vec{u}\in\mathbb{R}^N} \lambda J(\vec{u}) + \frac{1}{2}\|\vec{u} - \vec{g}\|^2. \tag{8}$$

Then, for all $z > 0$, the characteristic functions of the superlevel sets $E_z = \{u \geq z\}$ and $E'_z = \{u > z\}$ (which are different only if $z \in \{u_i, i = 1, \ldots, N\}$) are, respectively, the largest and smallest minimizer of

$$\min_{\vec{\theta}\in\{0,1\}^N} \lambda J(\vec{\theta}) + \sum_{i=1}^{N} \theta_i(z - g_i). \tag{9}$$

Since eq. (8) can be easily solved by Chambolle's algorithm, the solution of eq. (9) (binary problem) can be obtained by solving eq. (8) using Chambolle's algorithm followed by performing the thresholding to its output $\vec{u}$ with an appropriate threshold value z.

3.2 Transform of energy function in image segmentation

Let m_d be the difference between m_0 and m_1, and m_s be the addition of them, *i.e.* $m_d = m_0 - m_1$ and $m_s = m_0 + m_1$. By substituting them into eq. (1), we obtain

$$f(\vec{x}) = (1-\beta)m_d \sum_{u\in V} x_u\Big(\frac{m_s}{2} - y_u\Big) + \text{constant} + \beta \sum_{(u,v)\in E} |x_u - x_v|. \tag{10}$$

We define a new energy function by eliminating the constant term and dividing it by $(1-\beta)m_d$ as

$$f_{\mathrm{B}}(\vec{x}) = \frac{\beta'}{m_d} \sum_{(u,v)\in E} |x_u - x_v| + \sum_{u\in V} x_u\Big(\frac{m_s}{2} - y_u\Big), \tag{11}$$

where β' is $\beta/(1-\beta)$.

We note that the smoothing term in eq. (11) is the TV term. Therefore, from Proposition 1, the solution of eq. (11) can be obtained by thresholding the minimum solution of the following TV problem with the threshold value $m_s/2$.

$$f_{\mathrm{R}}(\vec{w}) = \frac{\beta'}{m_d} \sum_{(u,v)\in E} |w_u - w_v| + \frac{1}{2}\|\vec{w} - \vec{y}\|^2 \tag{12}$$

3.3 Proposed algorithms

Let us consider the case where we would like to segment an image for various combinations of (m_0, m_1) values simultaneously. From the discussion in Section 3.2., if the difference m_d is equal, their solutions can be obtained simply by changing the threshold value $m_s/2$ for the same minimum solution of the TV problem arg min $f_{\mathrm{R}}(\vec{w})$. Using this novel property, because the range of m_d is limited to $1, 2, \ldots, G-1$, we can obtain all the solutions correponding to all combinations of (m_0, m_1) values by the $G-1$ TV minimizations, *i.e.* the brute-force $G(G-1)/2$ graph cut minimizations are not necessary.

Algorithm 1 Proposed method for Scenario 1

Input: A measured image $\vec{y} \in \mathbb{R}^N$ and a smoothing parameter $\beta \in (0, 1)$
Output: A label image $\vec{x}^* \in \{0, 1\}^N$

$\beta' \leftarrow \beta/(1-\beta)$
for $m_d \leftarrow 1$ **to** $G-1$ **do**
  $\vec{w} \leftarrow$ Chambolle$(\vec{y}, \lambda \leftarrow \beta'/m_d)$
  for $m_0 \leftarrow 0$ **to** $G-1-m_d$ **do**
    $m_1 \leftarrow m_d + m_0$, $m_s \leftarrow m_0 + m_1$
    $\vec{x} \leftarrow$ Threshold$(\vec{w}, m_s/2)$
    $e_{m_0,m_1} \leftarrow f_{\mathrm{B}}(\vec{x}_{m_0,m_1})$
  end for
end for
$(m_0^*, m_1^*) \leftarrow \arg\min_{m_0,m_1} e_{m_0,m_1}$
return $\vec{x}^* = \vec{x}_{m_0^*,m_1^*}$

Therefore, Scenario 1 can be solved very efficiently by Algorithm 1. When we execute Chambolle's projection algorithm multiple times, we could further accelerate the computation by using the obtained dual solution $\vec{p}$ as an initial dual vector to solve the next problem with a different value of m_d. The exhaustive search, *i.e.* the method to apply the conventional method for all combinations of (m_0, m_1) values, requires $O(G^2)$ computation time, whereas the proposed algorithm requires $O(G)$ time, if the computation time of both minimizations, *i.e.* the graph cut and the TV minimization, is $O(1)$.

The method to solve Scenario 2 can be obtained from Algorithm 1 as follows. In Algorithm 1, we substituted the value of β'/m_d in the formulation of segmentation by the single regularization parameter λ when applying Chambolle's algorithm. Since λ is a constant when the ratio of β' and m_d is same, *i.e.* $\lambda = \beta'/m_d$, each $\vec{x}$ obtained by Algorithm 1 can be regarded as a common solution to multiple different values of β. The method to solve Scenario 2 is shown in Algorithm 2. It can reduce $O(nG^2)$ computation time to $O(G)$, where n is the number of β for which the user would like to obtain the segmentations. In implementation, in order to solve the lost correspondence between m_d and (m_0, m_1) arising due to the discretization, we approximate it by neighboring interpolated values.

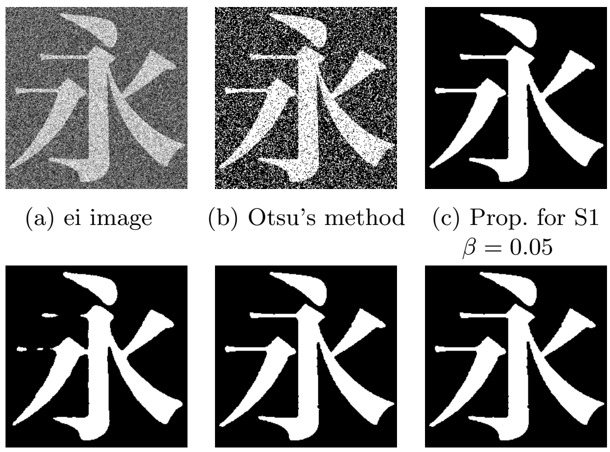
Algorithm 2 Proposed method for Scenario 2

Input: A measured image $\vec{y}$ and a set of smoothing parameters $\{\beta_k\}_{k=1}^n$ in ascending order
Output: Label images $\{\vec{x}_k^*\}_{k=1}^n$

for $k \leftarrow 1$ **to** n **do**
  $\beta'_k \leftarrow \beta_k/(1-\beta_k)$, $c_k \leftarrow \beta'_k/\beta'_1$
end for
for $m'_d \leftarrow 1$ **to** $G-1$ **do**
  $\vec{w} \leftarrow$ Chambolle$(\vec{y}, \lambda \leftarrow \beta'_1/m'_d)$
  for $m_s \leftarrow m'_d$ **to** $2(G-1)-m'_d$ **do**
    $\vec{x} \leftarrow$ Threshold$(\vec{w}, m_s/2)$
    for $k \leftarrow 1$ **to** n **do**
      for m_d **in** integers with $c_k m'_d$ as the nearest **do**
        $m_0 \leftarrow (m_s + m_d)/2$, $m_1 \leftarrow (m_s - m_d)/2$
        if $(m_0, m_1) \in$ range search **then**
          $e_{m_0,m_1,k} \leftarrow f_{\mathrm{B}}(\vec{x}_{m_0,m_1}, \beta \leftarrow \beta_k)$
        end if
      end for
    end for
  end for
end for
for $k \leftarrow 1$ **to** n **do**
  $(m_{0k}^*, m_{1k}^*) \leftarrow \arg\min_{m_0,m_1} e_{m_0,m_1,k}$
end for
return $\{\vec{x}_k^* = \vec{x}_{m_{0k}^*,m_{1k}^*}\}_{k=1}^n$

4 Experimental results

We implemented Otsu's method (with no smoothing term), the exhaustive minimization method (using Tayama's continuous optimization approach [1] to obtain each segmentation), and the proposed methods to various images, all of which consist of 256×256 pixels and 256 gray levels. All density values were normalized to $[0, 1]$. We used the Jaccard Index $\frac{|A \cap B|}{|A \cup B|}$ [8] to evaluate accuracy of segmentation. The implementation parameters of Chambolle's algorithm in the proposed method were selected as $\delta_t = 1/8$ and $tol = 1/64$ [4].



(a) ei image (b) Otsu's method (c) Prop. for S1 $\beta = 0.05$

(d) Prop. for S1 $\beta = 0.10$ (e) Exhaustive $\beta = 0.05, G = 32$ (f) Exhaustive $\beta = 0.05, G = 64$

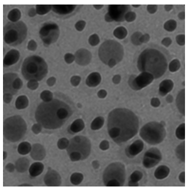
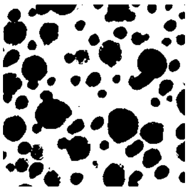
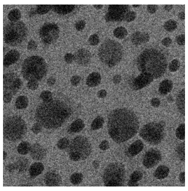
Figure 1: Experimental results for ei image.

We performed an experiment to evaluate validity and execution time of the algorithms for Scenario 1. Figure 1 and Table 1 show the experimental results for image "ei"(1a), which has density values $(m_0 = 100, m_1 = 200)$ and was degraded with independent Gaussian noise. Since the execution time of exhaustive method was extremely long, we implemented it by reducing the number of gray levels in the search space to 32 and 64 (from 256). It can be observed that the proposed method (1c, 1d) removed the effect of seg-

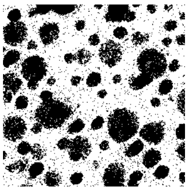
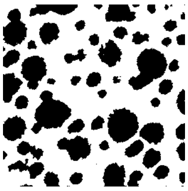
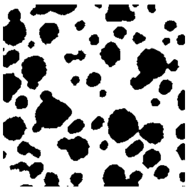
Table 1: Experimental results for ei image. J.I. denotes the Jaccard index with the ground truth. ET denotes the execution time.

	(m_0^*, m_1^*)	J.I.	ET (s)
Otsu (1b)	-	0.5128	0.13
Prop. for S1 (1c)	(100, 196)	0.9837	22.71
Prop. for S1 (1d)	(99, 195)	0.9300	20.56
Exhaustive (1e)	(104, 192)	0.9855	668.40
Exhaustive (1f)	(100, 196)	0.9856	2866.60

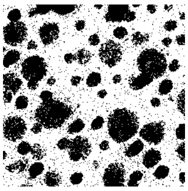
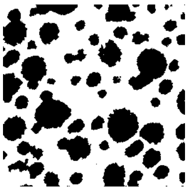
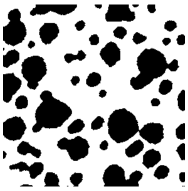
mentation errors due to the noise compared with the result of Otsu's method (1b). Although the results of exhaustive method (1e, 1f) are almost same as that of the proposed method (1c), its execution time was very long. Considering that the complexity of exhaustive method is $O(G^2)$, its computation time operated with $G = 256$ is likely to be 4.6×10^4(s), which is about 13 hours.

(a) fept image (b) fept, Otsu (c) Noised image

(d) Prop. for S1 $\beta = 0.0020$ (e) Prop. for S1 $\beta = 0.0079$ (f) Prop. for S1 $\beta = 0.0158$

(g) Prop. for S2 $\beta = 0.0020$ (h) Prop. for S2 $\beta = 0.0079$ (i) Prop. for S2 $\beta = 0.0158$

Figure 2: Experimental results for fept image.

In practical use of image segmentation, it is necessary to select an appropriate value of β in some way. Here, we performed an experiment of Scenario 2, in which multiple segmentation results with a number of different β are computed and the user finally selects the best one. We compared the proposed efficient Algorithm 2 and the exhaustive method to repeat Algorithm 1 for Scenario 1 with many candidate values of β. Figure 2 and Table 2 show the experimental results for image "fept" (2a), which is a transmission electron microscope (TEM) image of iron-platinum (Fe-Pt) nanoparticles. The corresponding noised image is shown Figure 2c. To simplify the implementation, candidate values of β_k $(k = 1, 2, \ldots, n)$ were given so that β'_k are integer multiples of β'_1, *i.e.* $\beta'_k = k\beta'_1$. We obtained accurate segmentations from the degraded image by selecting an appropriate parameter value

Table 2: Experimental results for fept image. J.I. denotes Jaccard index with the result of Otsu's method for non-degraded image (2b) as the ground truth.

		Prop. for S1		Prop. for S2	
β_k	β'_k	(m_0^*, m_1^*)	J.I.	(m_0^*, m_1^*)	J.I.
0.0020	0.002	(48, 109)	0.8522	(48, 109)	0.8522
0.0040	0.004	(49, 107)	0.9120	(49, 107)	0.9122
0.0060	0.006	(51, 106)	0.9328	(50, 106)	0.9308
0.0079	0.008	(51, 106)	0.9340	(51, 106)	0.9344
0.0099	0.010	(51, 106)	0.9313	(51, 106)	0.9323
0.0119	0.012	(50, 105)	0.9216	(51, 105)	0.9254
0.0138	0.014	(51, 106)	0.9226	(52, 105)	0.9250
0.0158	0.016	(52, 105)	0.9172	(52, 105)	0.9199

$\beta = 0.0079$ in the both compared methods. Comparing the two methods, the segmentation results corresponding to the same β value showed a high similarity. On the other hand, the total execution time was 126.41(s) for the exhaustive method and 53.71(s) for Algorithm 2, which demonstrates that Algorithm 2 is more efficient.

5 Conclusion

In this paper, we proposed an image segmentation method, which does not require inputting density parameter values (m_0, m_1), under the assumption that b.g. and f.g. have uniform densities degraded with additive noise (Algorithm 1). Furthermore, we also proposed a method to efficiently generate multiple segmentation results corresponding to a number of candidate β values simultaneously (Algorithm 2).

References

[1] T. Tayama, H. Kudo, H. Takizawa: "Exact solution method for MRF model-based image segmentation based on continuous optimization," *IEICE Technical Report (MI)*, vol. 113, pp. 13–18, 2014. (In Japanese)

[2] H. Ishikawa, "Graph Cut," *IPSJ SIG technical reports (CVIM)*, vol.158, no.26, pp.193–204, 2007. (In Japanese)

[3] A. Chambolle, J. Darbon: "On total variation minimization and surface evolution using parametric maximum flows," *IJCV*, vol. 84, pp. 288–307, 2009.

[4] J. Duran, B. Coll, and C. Sbert: "Chambolle's projection algorithm for total variation denoising," *IPOL*, vol. 3, pp. 311–331, 2013.

[5] A. Chambolle, "Total variation minimization and a class of binary MRF models," *EMMCVPR 2005*, LNCS, vol. 3757, pp. 136–152, 2005.

[6] Y. Boykov and M. P. Jolly: "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *ICCV*, vol. I, pp. 105–112, 2001.

[7] E. Zagrouba, S. Gamra, A. Najjar: "Model-based graph-cut method for automatic flower segmentation with spatial constraints," *Image Vis. Comput.*, vol. 32, pp. 1007–1020, 2014.

[8] K. Ogawa, H. Kudo, A. Shimizu *Ed.*: "Handbook of Medical Imaging Technology," p.437, 2012. (In Japanese)

[9] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys., Man., Cyber.*, vol. 9, no. 1, pp. 62–66, 1979.