An Efficient and Effective Tool for Image Segmentation, Total Variations and Regularization

Dorit S. Hochbaum⋆

Department of Industrial Engineering and Operations Research, University of
California, Berkeley
hochbaum@ieor.berkeley.edu

Abstract. One of the classical optimization models for image segmentation is the well known Markov Random Fields (MRF) model. MRF formulates many total variation and other optimization criteria used in image segmentation. In spite of the presence of MRF in the literature, the dominant perception has been that the model is not effective for image segmentation. We show here that the reason for the non-effectiveness is not due to the power of the model. Rather it is due to the lack of access to the optimal solution. Instead of solving optimally, heuristics have been engaged. Those heuristic methods cannot guarantee the quality of the solution nor the running time of the algorithm.

We describe here an implementation of a very efficient polynomial time algorithm, which is provably fastest possible, delivering the optimal solution to the MRF problem, Hochbaum (2001). It is demonstrated here that many continuous models, common in image segmentation, have a discrete analogs to various special cases of MRF. As such they are solved optimally and efficiently, rather than with the use of continuous techniques such as PDE methods that can only guarantee convergence to a local minimum.

The MRF algorithm is enhanced here demonstrating that the set of labels can be any discrete set. Other enhancements include dynamic features that permit adjustments to the input parameters and solves optimally for these changes with minimal computation time. Modifications in the set of labels (colors), for instance, are executed instantaneously. Several theoretical results on the properties of the algorithm are proved here and are demonstrated for examples in the context of medical and biological imaging.

1 Introduction

Partitioning and grouping of similar objects plays a fundamental role in image segmentation and in clustering problems. In such problems the goals are to group together similar objects, or pixels in the case of image processing. Given an input image, the objective of image segmentation is to recognize the salient features

in the image. Each feature set is grouped together in one segment represented by some uniform color area.

A noisy or corrupted image is characterized by lacking uniform color areas, which are assumed to characterize a true image. Rather, in such image there are adjacent pixels of different color areas. To achieve higher degree of uniform color areas, it is reasonable to assign a penalty to neighboring pixels that have different colors associated with them. On the other hand, the purpose of the segmentation is to represent the "true" image. For that purpose the given assignment of colors in the input image is considered to be the "priors" on the colors of the pixels, and as such, the best estimate available on their true labels. Therefore, any change in those priors is assigned a penalty for deviating from the priors.

The Markov Random Fields problem for image segmentation is to assign colors to the pixels so that the total penalty is minimized. The penalty consists of two terms. One is the *separation* penalty, or *smoothing* term, and the second is the *deviation* penalty, or *fidelity* term. For this reason we refer to this penalty minimization problem also as the *separation-deviation* problem. This problem has been extensively studied over the past two decades, see e.g. [3], [5], [11], [12], [16], [17]. The problem formulation, described in full detail in Section 3 is

$$\text{(MRF)} \quad \min \quad \sum_{i \in V} G_i(x_i) + \sum_{i \in P} \sum_{j \in N(i)} F_{ij}(x_i - x_j)$$
$$\text{subject to} \quad x_i \in X \quad \forall \, i \in P.$$

It is noted that the concept of "colors" associated with pixels can be replaced by any other scalar characterization of pixels or voxels, such as texture. We refer here to colors as a representation of such characterizations.

The complexity of MRF depends on the form of the penalty functions. A full classification of the problem's complexity is given in [15] showing that for convex penalty functions the problem is polynomially solvable, and for non-convex the problem is NP-hard. The cases when the deviation penalty functions are convex and the separation penalty functions are linear was shown by Hochbaum [15] to be solvable in polynomial time using a parametric cut procedure. Furthermore, it was shown that the complexity of the algorithm is the fastest possible. The case when both separation and deviation penalty functions are convex were also shown to be solvable very efficiently by [1,2]. For non-convex penalty functions the MRF problem is NP-hard.

Problems of total variations and regularization have been utilized in image analysis for the purpose of *denoising* an image. These employ continuous methodologies. Recent works that provide approximate methods for solving MRF utilize convex relaxations (e.g. Pock et al. [21]) along with primal-dual approaches, may not converge to an optimal solution, and the running time cannot be determined in advance. This is surprising, given that the exact discrete problem can be solved within guaranteed polynomial time complexity. Moreover, digital images are inherently discrete, and considering them as continuous causes loss of accuracy. The output of a continuous method must be mapped back to digital image information, entailing further loss of accuracy. We demonstrate that several classical continuous models are better represented with the MRF model and thus benefit from the algorithmic efficiency of solving it.

2   Relationship to Continuous Models

In the total variation method [19,23] the recorded image is represented by the function which maps each pixel to its label (color). It is assumed that u_0 can be decomposed as $u_0 = u + v$ where u contains homogeneous regions with sharp prominent edges, and v contains additional texture and noise. The goal of the total variation method is to find u by minimizing the functional

$$\int_\Omega |\nabla u| dx dy + \alpha ||u - u_0||.$$

This functional is define on the plane, where (x, y) designate the position of each pixel in the image.

Although not immediately apparent, there is a connection between this problem and the MRF problem: The term $|\nabla u|$ captures the difference between each pixel and its neighborhood. The neighborhood can be set to any desirable set – it is not restricted to the commonly used grid neighborhood. This gradient term is thus the *separation* term. The second term $\alpha ||u - u_0||$ is the deviation of the mapped function u from the recorded image u_0.

This total variation problem is solved by continuous techniques. One such method solves the associated Euler-Lagrange equation

$$u = u_0 + \frac{1}{2\alpha} \nabla \cdot (\frac{\nabla u}{||\nabla u||}).$$

In contrast to MRF, this method does not guarantee to deliver an optimal solution and its complexity is undetermined. For this problem MRF does deliver an optimal solution to this problem, and in polynomial time.

In a more general set-up, the total variation regularization problem (TVR) the image is represented as $s(x)$ – a given function define on an open subset Ω, and $f(x)$ is its *regularized* version, or for images, it is called the *denoising* of s. We define two real functions $\gamma : R \to [0, \infty)$ and $\beta : R \to [0, \infty)$ which assume the value 0 for the argument of 0,

$$F(f) = \int_\Omega \gamma(f(x) - s(x)) dx$$

In the denoising literature F is called a *fidelity* term since it measures deviation from $s()$ which could be a noisy grayscale image. In our terminology, the fidelity term is the *deviation*.

A second function is the total variations on f, $TV(f)$: The discrete form of the total variations function is represented as a function f on a grid of discrete values in Ω and associated with a defined *neighborhood* of each grid point. Let the set of neighboring pairs be denoted by E. Then the total variation of f is $\sum_{[i,j]\in E} \beta(f(i) - f(j))$ for a function β often selected as the absolute value function: $\beta(x) = \max\{0, x\}$. For a constant α the *total variation regularization* of $s()$ is the function f that minimizes the weighted combination of the total variations and fidelity of f:

$$\min TV(f) + \alpha F(f)$$

Rudin, Osher and Fatemi [23] have studied TVRs of F where $\gamma(y) = y^2$, and Chan and Esedoglu [7] studied $\gamma(y) = |y|$.

Since MRF is solved in polynomial time for convex γ and convex β, consequently, the problem of Chan and Esedoglu is a special case solved by parametric cut, and the problem of Rudin et al. is a special case solved by the quadratic convex dual of min cost network flow. Both cases are efficiently solvable and the MRF algorithm guarantees an optimal solution in polynomial time.

The MRF problem can also be used to represent certain classes of the Mumford-Shah problem, as well as several image analysis problems that are addressed with the eigenvector technique. The details of these mappings are to be described in the full version of this paper.

3   The Methodology

For the MRF model for the image segmentation problem the input is an image constituting of a set of pixels each with a given color and a neighborhood relation between pairs of pixels. The decision is to assign each pixel a color assignment, that may be different from the given color of the pixel, so that neighboring pixels will tend to have the same color assignment. The aim is to modify the given color values as little as possible while penalizing changes in color between neighboring pixels. The penalty function thus has two components: the deviation cost that accounts for modifying the color assignment of each pixel, and the separation cost that penalizes the extent of pairwise discontinuities in color assignment for each pair of neighboring pixels.

Formally, we are given a graph $G = (V, A)$, or an image which is a set of pixels V, with a real-valued intensity r_i for each pixel $i \in V$. The neighborhood of pixel i, which contains pixels adjacent to i, is denoted by $N(i)$. The set of pairs of nodes and their neighbors is denoted by A. So $A = \{(i, j)|j \in N(i)\}$. Note that for every pair of neighbors $\{i, j\}$ the graph G contains two arcs $(i, j).(j, i) \in A$. We wish to assign each pixel $i \in V$ an intensity x_i that belongs to a discrete finite set $X = \{i_1, i_2, \ldots, i_k\}$ so that the sum over all pixels of the *deviation* cost $G_i(\cdot)$ and the *separation* cost $F_{ij}(\cdot)$ is minimized. Note that the values of x_i do not have to be selected from the same set as the value of r_i, as shown here for the first time in Lemma 1. The deviation function depends on the deviation of the assigned color from the given intensity $G_i(x_i - r_i)$. The separation is a function of the difference in assigned intensities between adjacent pixels $F_{ij}(x_i - x_j)$. The problem is stated as follows.

$$\min \quad \sum_{i \in V} G_i(x_i) + \sum_{i \in V} \sum_{j \in N(i)} F_{ij}(x_i - x_j)$$
$$\text{subject to} \quad x_i \in X \quad \forall\, i \in V.$$

We refer to the special case of the MRF problem with each variable x_i taking an integer value in an interval $[\ell_i, u_i]$ as the separation-deviation problem.

The separation-deviation problem was shown in [1,2,15] to be solvable in polynomial time when the functions $G_i(\cdot)$ and $F_{ij}(\cdot)$ are convex. Note that when those functions are not convex the problem is NP-hard, although when only the

functions $G_i(\cdot)$ are nonlinear, and $F_{ij}(\cdot)$ are convex the problem is solved in pseudopolynomial time with run time that depends on the number of values of X, k, [1]. The important case we will focus on here is with $G_i(\cdot)$ convex and $F_{ij}(x_i - x_j)$ *bi-linear* forming a two piecewise linear function which is linear in the range $x_i \geq x_j$ and linear in the range $x_j \geq x_i$. For constants u_{ij}, u_{ji} the function is defined as:

$$F_{ij}(x_i - x_j) = \begin{cases} u_{ij} & \text{if } x_i > x_j \\ 0 & \text{if } x_i = x_j \\ u_{ji} & \text{if } x_i < x_j. \end{cases}$$

For convex functions $G_i(\cdot)$ and bi-linear functions $F_{ij}(\cdot)$, the formulation is equivalent to the following constrained optimization problem, referred to as (SD) (standing for Separation-Deviation):

$$
\begin{aligned}
\text{(SD) min} \quad & \sum_{j \in V} G_j(r_j, x_j) + \sum_{(i,j) \in A} F_{ij}(z_{ij}) \\
\text{subject to} \quad & x_i - x_j \leq z_{ij} \quad \text{for } (i,j) \in A \\
& u_j \geq x_j \geq \ell_j \quad j = 1, \ldots, n \\
& z_{ij} \geq 0 \quad (i,j) \in A.
\end{aligned}
$$

The complexity of this problem was shown in [15] to be $O(T(n,m) + n \log U)$ where $T(n,m)$ is the complexity of solving the minimum cut problem on a graph with n nodes and m arcs and U is the length of the interval for the color values – the number of possible labels – or as we show here, $|X|$. For the formulation above $U = \max_j\{u_j - \ell_j\}$. The second complexity term is required to find the minima of convex functions. In all our implementations the convex functions are piecewise linear (e.g. absolute value function) or quadratic. In those cases the second term vanishes and the complexity of the procedure is $T(n,m)$. The algorithm used solves the (SD) problem for any size of color set as a parametric minimum cut problem, in the complexity of a single minimum cut procedure. The algorithm used to solve the parametric minimum cut problem is the pseudoflow algorithm of [14], for which the software is available to download at [8]. The complexity of this algorithm was shown recently in [13] to be $T(n,m) = O(mn \log \frac{n^2}{m})$.

We show next that the algorithm solving (SD) extends to the MRF problem with $x_i \in X$ for any set of discrete values X. We first review the algorithm of [15] and then prove, in Lemma 1, that it extends to the MRF problem with an arbitrary discrete set of feasible values.

We define an s,t-graph $G_\alpha = (V_{st}, A_{st})$ from the adjacency graph of the image (V, A) where V is the set of pixels and A the set of adjacency arcs. For $\ell = \min_j \ell_j$ and $u = \max_j u_j$, we choose a parameter value $\alpha \in (\ell, u)$. For each arc (i,j) the arc capacity is u_{ij}.

We add to the set of nodes V a source s and sink t, $V_{st} = V \cup \{s,t\}$. Next let $G_i'(\alpha)$ be the subgradient of $G_i()$ at α, $G_i(\alpha) - G_i(\alpha - 1)$. Let the subgradient value of function $G_i(x)$ to be equal to M at values of $x > u_i$, and to $-M$ for values $x < \ell_i$, for M a suitably large value. With this extension the box constraints are uniform for all variables, $u \geq x_j \geq \ell$ we replace the weights of the nodes and set, for each node $v \in V$, by an arc adjacent to the source of capacity $c_{sv} = \max\{0, G_v'(\alpha)\}$, and an arc adjacent to the sink t of capacity
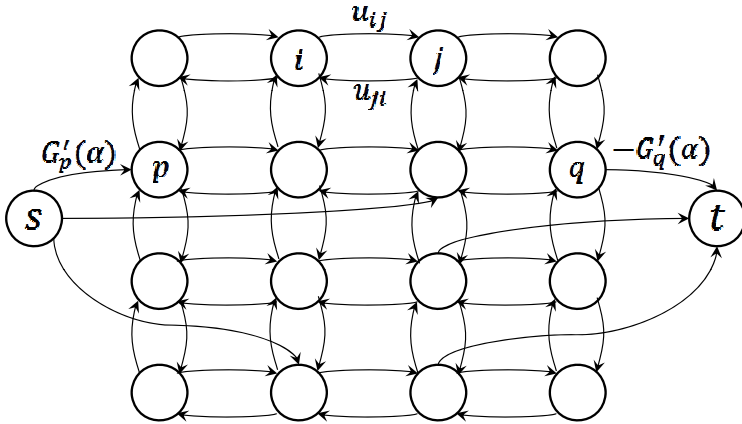
Fig. 1. The graph G_α

$c_{vt} = \max\{0, -G'_v(\alpha)\}$. Let the set of arcs of positive capacity adjacent to the source be denoted by A_s, and the set of arcs of positive capacity adjacent to the sink, A_t. The remainder of the arcs, for each arc (i,j), $j \in N(i)$ have capacities u_{ij}. Let the minimum cut $(\{s\} \cup S, \bar{S} \cup \{t\})$ in the graph G_α partition V to $S = S_\alpha$ and $V \setminus S = \bar{S}_\alpha$. The graph G_α is illustrated in Fig. 1 for an example of a grid graph (V, A) describing the adjacencies. Note however that the algorithm described works for any type of graph, rather than for grid graphs only.

Let the optimal solution to (SD) be $\mathbf{x}^* = (x_j^*)$. The key to the efficient algorithm to the (SD) problem is the threshold theorem:

Theorem 1 (The threshold theorem [15]). *The optimal solution $\mathbf{x}^*$ to (SD) satisfies $x_j^* < \alpha$ for all $j \in S_\alpha$, and $x_j^* \geq \alpha$ for all $j \in \bar{S}_\alpha$.*

The threshold theorem means that for each node we can determine whether the corresponding variable's value in an optimal solution is $< \alpha$ or $\geq \alpha$, depending on whether the respective node belongs to the source or the sink set of the cut. See Fig. 2 for illustration.

By solving for each value of α in the range, the threshold theorem can be used to establish a partition of the nodes in the graph, and the corresponding variables, to sets where in each set all variables get the same value (and same color) in an optimal solution.

Instead of solving for each value of α we find all the *breakpoints* where the cut set is changing. Let S_{λ_q} be the *minimal* source set obtained by solving the minimum cut problem in the graph corresponding to parameter λ_q. Then, for a sequence of monotone increasing values of the parameter, $\lambda_0 < \lambda_1 < \lambda_2 \ldots < \lambda_p$, we get a *nested* collection of source sets of the respective minimum cuts: $\{s\} = S_{\lambda_1} \subset S_{\lambda_2} \subset \ldots \subset S_{\lambda_p} \subset V$. See Fig 3 for illustration. When $\lambda_0 \leq \ell$ then the set of nodes of value $< \lambda_0$ is empty. For $\lambda_p \geq u$ the set of nodes
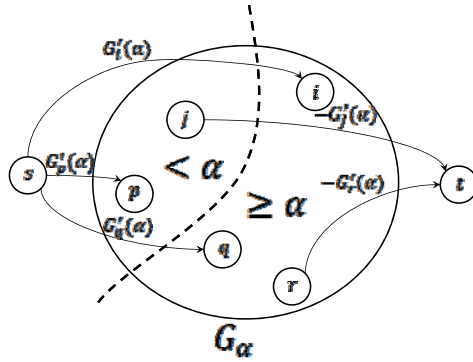
Fig. 2. The threshold theorem: The dashed line represents the arcs of the cut

of value $< \lambda_p$ is V. Therefore, in the optimal corrected image, all pixels in $S_q = S_{\lambda_q} \setminus S_{\lambda_{q-1}}, q = 2, 3, \ldots, p$ have intensity strictly less than λ_q and greater or equal than λ_{q-1}.

Notice that it is sufficient to generate the values of the breakpoints as integers. That is because the values of the variables determined in each set of the partition can take only integer values, so the smallest integer value in the interval $[\lambda_{q-1}, \lambda_q)$ will be the value assigned to all nodes/variables in the set S_q. Hence the values of the breakpoints λ_i do not need to be contained in the set X. However, we will let the set X consist of labels that are integer values.

Since the source set does not change for any $\alpha \in [\lambda_{q-1}, \lambda_q)$, we conclude that for all $j \in S_q$ x_j^* is equal to the smallest value in X that is $\geq \lambda_{q-1}$.

Consider the extension of (SD) to (SD'):

$$\text{(SD') min} \quad \sum_{i \in V} G_i(x_i) + \sum_{i \in V} \sum_{j \in N(i)} F_{ij}(x_i - x_j)$$
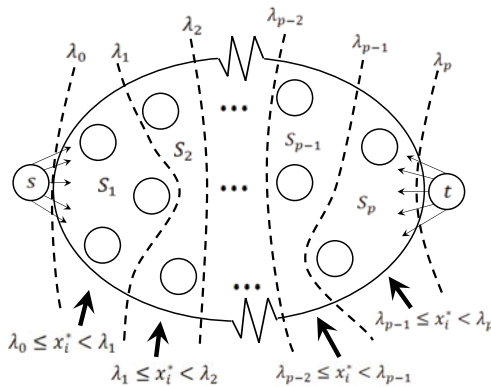$$\text{subject to} \quad x_i \in X \quad \forall \, i \in V.$$

Fig. 3. The parametric cut

Lemma 1. *Given the set of integer breakpoints $\lambda_0 < \lambda_1 < \lambda_2 \ldots < \lambda_p$, the optimal solution to (SD') restricted to any set of colors is generated in linear time.*

Proof: The proof is constructive. Let $X = \{i_1, \ldots, i_k\}$. Let V be the set of all the pixels/nodes.

For i_1 let λ_{ℓ_1} be the largest breakpoint smaller or equal to i_1, $\lambda_{\ell_1} = \arg\max \lambda_{\ell_j} \leq i_1$. Assign to all variables with nodes in $S_1 \cup \ldots \cup S_{\ell_1}$ the value i_1. Update $V \leftarrow V \setminus \{S_1 \cup \ldots \cup S_{\ell_1}\}$. Let i_ℓ be the largest value in X less than $\lambda_{\ell_1 + 1}$. Update $X \leftarrow X \setminus \{i_1, \ldots, i_\ell\}$.

The following iterative step is repeated until all variables values have been assigned and $V = \emptyset$.

Iterative step:

Let i_q be the first (smallest) value in X. Then $i_q \geq \lambda_{\ell_1 + 1}$. Let λ_{ℓ_q} be the largest breakpoint smaller or equal to i_q, $\lambda_{\ell_q} = \arg\max \lambda_{\ell_j} \leq i_q$. Assign to all variables with nodes in $S_q \cup \ldots \cup S_{\ell_q}$ the value i_q. Update $V \leftarrow V \setminus \{S_q \cup \ldots \cup S_{\ell_q}\}$. Let i_ℓ be the largest value in X less than $\lambda_{\ell_q + 1}$. Update $X \leftarrow X \setminus \{i_q, \ldots, i_\ell\}$.

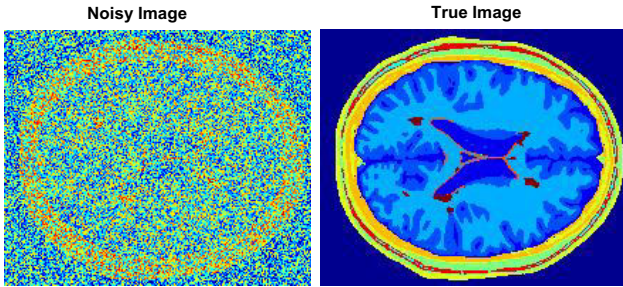The correctness of the procedure follows from the threshold theorem.   □

Fig. 4. Brain image 1, true and noisy

4   Experimental Results

4.1   Denoising by Modifying the Ratio between the Separation and Deviation Penalties

The implementation solves the MRF problem with parametric coefficients S and D multiplying the respective terms of separation and deviation. Note that only changes in the ratio $\frac{S}{D}$ have an effect on the optimal solution, rather than the actual values of S and D.

$$(MRF) \min \quad D \sum_{j \in V} G_j(r_j - x_j) + S \sum_{(i,j) \in A} u_{ij}|x_i - x_j|$$
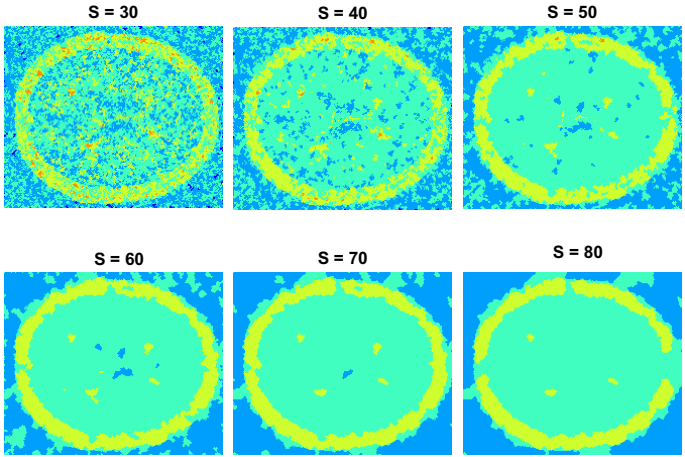$$\text{subject to} \quad x_i \in X \quad \text{for } i \in V .$$

Fig. 5. The output for increasing values of S when applied to noisy brain image 1

The effect of modifying the ratio $\frac{S}{D}$ is illustrated here for two examples of brain images. The first set of true and noisy images are given in Fig. 4. In that image there are four small lesions. We then apply the separation-deviation algorithm with $D = 2$ and for increasing values of S, as shown in Fig. 5. The lesions show very clearly in the high separation (S values of 60 or 70) images in yellow color.

4.2   Increasing Deviation for a Selected Color

The algorithmic tool allows to select a particular color, either by the color code, or by clicking on a pixel that has the desired color. The deviation penalty is then increased for all integer color codes in a small interval around the selected color. For color code q the interval is $[q - 5, q + 5]$. The size of this interval can be adjusted by the user.
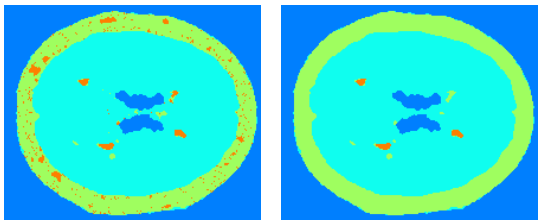

Fig. 6. Increased deviation penalty for a selected color in brain image 1

We show here, for brain image 1, that if the color orange is selected, then it shows as the color of 3 out of the 4 lesions, see Fig. 6 above. When the deviation for that color is increased the lesions become better segmented and more prominent. Of course, the color orange also appears in other areas of the brain shell where it is of no clinical significance. This issue will be addressed in the next prototype of the interactive tool, where the deviation increase will apply only in a user-defined window.

4.3   Comparison of Image Segmentation with Separation-deviation to the Normalized Cut Approach

We now compare our software for image segmentation with the normalized cut approach introduced by Shi and Malik, [25]. This normalized cut approach utilizes the spectral technique in finding the Fielder eigenvalue and the corresponding eigenvector. The method is described and Shi's software implementation is provided in: `http://www.cis.upenn.edu/~jshi/software/`

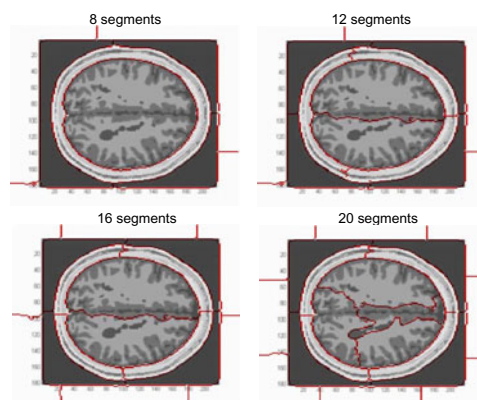The input to that code is the number of desired segments in the output image.

Fig. 7. Normalized cut software segmentation of true brain image 2 for 8, 12, 16 and 20 segments

The code preprocesses the input image, first by converting it to gray scale and then resizing it to 160×160. The algorithm is then applied to the the preprocessed image.

We show here the segmentation of a brain image, brain image 2. This is shown in Fig. 7. Only the 20 segments begins to show the lesion area, but still does not delineate it correctly. This is compared in Fig. 8 to the segmented and traced lesions found with the solution of (SD) applied to the same image. (The software of Shi requires to convert the image first to gray scale, which is why it is not presented in color.)
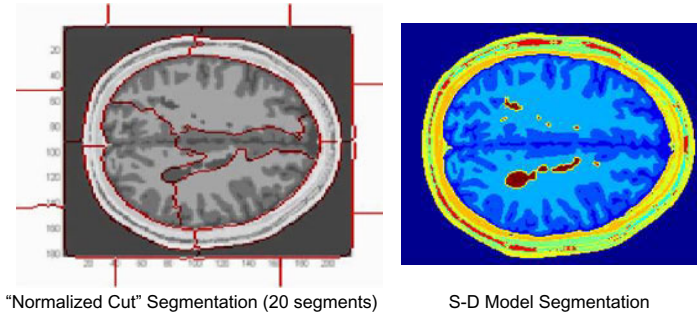
"Normalized Cut" Segmentation (20 segments)   S-D Model Segmentation

Fig. 8. Comparison of the normalized cut software segmentation and the (SD) segmentation of the true brain image 2

5   Conclusions

We demonstrate here that the MRF algorithm is an effective technique for regularization and denoising of images, in theory and in practice. Since the algorithm delivers an optimal solution, and is provably fastest possible, it gives better quality results than any alternative methodology, in terms of minimizing the objective function. The algorithm is shown here to segment successfully the salient features in true images, and to be able to identify hidden important features and de-blur noisy images. These capabilities make the algorithm a useful addition to a segmentation tool box.

References

1. Ahuja, R.K., Hochbaum, D.S., Orlin, J.B.: A cut-based algorithm for the convex dual of the minimum cost network flow problem. Algorithmica 39(3), 189–208 (2004)
2. Ahuja, R.K., Hochbaum, D.S., Orlin, J.B.: Solving the convex cost integer dual network flow problem. Management Science 7, 950–964 (2003)
3. Blake, A., Zisserman, A.: Visual reconstruction. MIT Press, Cambridge (1987)
4. Boykov, Y., Jolly, M.-P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: International Conference on Computer Vision (ICCV), vol. I, pp. 105–112 (2001)
5. Boykov, Y., Veksle, O., Zabih, R.: Markov random fields with efficient approximations. In: Proc IEEE Conference CVPR, Santa Barbara, CA, pp. 648–655 (1998)
6. Boykov, Y., Veksle, O., Zabih, R.: Fast approximate energy minimization via graph cuts. In: Proc 7th IEEE International Conference on Computer Vision, pp. 377–384 (1999)
7. Chan, T.F., Esedoglu, S.: Aspects of total variation regularized l1 function approximation. SIAM J. on Applied Math. 65(5), 1817–1837 (2005)
8. Chandran, B.G., Hochbaum, D.S.: Pseudoflow solver (accessed, January 2007), http://riot.ieor.berkeley.edu/riot/Applications/Pseudoflow/maxflow.html

9. Collins, D.L., Zijdenbos, A.P., Kollokian, V., Sled, J.G., Kabani, N.J., Holmes, C.J., Evans, A.C.: Design and construction of a realistic digital brain phantom. IEEE Transactions on Medical Imaging 17(3), 463–468 (1998)
10. Cox, I.J., Rao, S.B., Zhong, Y.: Ratio regions: A technique for image segmentation. In: Proc. Int. Conf. on Pattern Recognition. B, pp. 557–564 (1996)
11. Geiger, D., Girosi, F.: Parallel and deterministic algorithms for MRFs: surface reconstruction. IEEE Transactions on Pattern Analysis and Machine Interlligence 13, 401–412 (1991)
12. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 6, 721–741 (1984)
13. Hochbaum D. S., Orlin J. B.: Pseudoflow algorithm in $O(mn \log n^2/m)$ time, UC Berkeley (manuscript) (submitted, 2007)
14. Hochbaum, D.S.: The Pseudoflow algorithm: A new algorithm for the maximum flow problem. Operations Research 4, 992–1009 (2008)
15. Hochbaum, D.S.: An efficient algorithm for image segmentation, Markov random fields and related problems. Journal of the ACM 4, 686–701 (2001)
16. Ishikawa, H., Geiger, D.: Segmentation by grouping junctions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1998, pp. 125–131 (1998)
17. Li, S.Z., Chan, K.L., Wang, H.: Bayesian image restoration and segmentation by constrained optimization. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1996 (1996)
18. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Int. J. Comp. Vision 43, 7–27 (2001)
19. Osher, S.J., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Springer, New York (2003)
20. Pham, D.L., Xu, C., Prince, J.L.: A survey of current methods in medical image segmentation. Annual Review of Biomedical Engineering 2, 315–337 (2000)
21. Pock, T., Chambolle, A., Cremers, D., Bischof, H.: A convex relaxation approach for computing minimal partitions. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 810–817 (2009)
22. Pretorius, P.H., King, M.A., Tsui, B.M.W., LaCroix, K.J., Xia, W.: A mathematical model of motion of the heart for use in generating source and attenuation maps for simulating emission imaging. Med. Phys. 26, 2323–2332 (1999)
23. Rudin, L.I., Osher, S.J., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D 60, 259–268 (1992)
24. Sarkar, S., Boyer, K.L.: Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, p. 478 (1996)
25. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 888–905 (2000)
26. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt, A.: Hierarchy and adaptivity in segmenting visual scenes. Nature 442, 810–813 (2006)
27. Wang, S., Siskind, J.M.: Image segmentation with ratio cut. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 25(6), 675–690 (2003)
28. Tolliver, D.A., Miller, G.L.: Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In: CVPR 2006, pp. 1053–1060 (2006)