COLOR IMAGE SEGMENTATION USING ITERATIVE EDGE CUTTING, NUV-EM, AND GAUSSIAN MESSAGE PASSING

Boxiao Ma, Nour Zalmai, Róbert Torfason, Carina Stritt[†]*, and Hans-Andrea Loeliger*

ETH Zurich, Dept. of Information Technology & Electrical Engineering
[†]EMPA, Dübendorf, Switzerland, Center for X-ray Analytics
{ma, zalmai, loeliger}@isi.ee.ethz.ch, robertto@student.ethz.ch, carina.stritt@empa.ch

ABSTRACT

A new approach to image segmentation (grayscale or color) is proposed. It uses a (improper) Markov random field prior with sparsifying NUV terms (normal with unknown variance), which favors piecewise smooth images with sharp edges. The proposed algorithm iterates two steps. In the first step, the unknown scalar variances are learned by approximate EM (expectation maximization). The actual computations for this step boil down to iterative scalar Gaussian message passing, which scales linearly with the number of pixels. In the second step, all edges that were detected in the first step are cut and removed from further processing. Simulation results demonstrate that the proposed approach compares favorably with four standard methods.

Index Terms— Image segmentation, sparse Bayesian learning, expectation maximization, Gaussian message passing

1. INTRODUCTION

Image segmentation is the process of partitioning an image into non-overlapping segments so that pixels belonging to the same segment share certain properties. Applications include machine vision, medical image analysis, object detection and recognition, etc. Image segmentation is known to be one of the most important and difficult tasks in image processing.

A great variety of segmentation algorithms exists, based on techniques such as thresholding, clustering, non-linear transformations, edge detection, and many others [1–6]. Many of these algorithms rely on edge detection, which is known to be sensitive to parameter settings and requires additional efforts to close regions [3]. Also, many segmentation algorithms are based on graph structures, often using Markov random fields (MRFs), which usually model the image as an undirected graph with edge weights defined by some (dis)similarity measure between neighboring pixels [7–9].

In this paper, we use both edge detection and a specific (improper) MRF prior. The latter was recently proposed in [10] and uses normal random variables with unknown variances (NUV) from sparse Bayesian learning [11–13]. (A related, but different, MRF prior was proposed in [14].) The proposed algorithm, which differs substantially from other segmentation algorithms, is iterative, and each iteration consists of two steps. In the first step, unknown parameters (variances) of the MRF prior are learned by an approximate EM algorithm. The pertinent computations boil down to an inner loop of iterative scalar Gaussian message passing, which scales linearly with the number of pixels. An edge (between two neighboring pixels) is detected whenever the corresponding variance is nonzero. In the second step (which is computationally trivial), all detected edges are cut and removed from further processing.

The proposed algorithm is empirical: we do not claim it to converge to the global minimum of some plausible cost function. However, the algorithm is based on solid concepts (NUV priors from sparse Bayesian learning [11–13] and iterative scalar Gaussian message passing [15–17]), and it works very well in practice. According to experimental results, our approach yields arguably better segmentations than some standard methods.

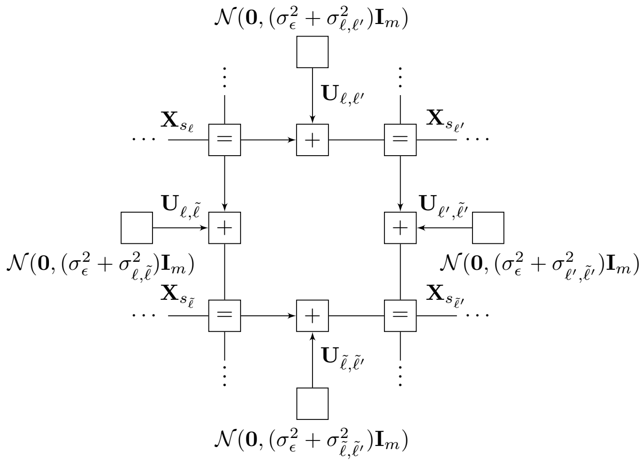
The paper is structured as follows. In Section 2, we define the image representation. The NUV prior and the edge detection algorithm are described in Section 3. Edge cutting is described in Section 4, and some experimental results are reported in Section 5.

2. IMAGE REPRESENTATION

We assume that the pixels are arranged in a 2-dimensional rectangular grid. Each pixel has a unique spatial index $s_\ell = (i, j)$ with $\ell \in \{1, \ldots, L\}$, where i is the row index, j is the column index, and L is the total number of pixels.

Let $B \subset \{1, \ldots, L\}^2$ be the set of nearest-neighbor pairs: for any two pixels at locations $s_\ell = (i, j)$ and $s_{\ell'} = (i', j')$, $(\ell, \ell') \in B$ if and only if both $|i-i'|+|j-j'| = 1$ and $\ell < \ell'$.

The color (or grayscale value) of each pixel is a vector (or scalar) $\mathbf{y}_{s_\ell} \in \mathbb{R}^m$ with $m = 3$ for color images and $m = 1$ for grayscale images. For color images, we will not use the standard RGB representation, but the color space CIE $L^*a^*b^*$ [18], which offers a better match between human visual perception and Euclidean distance. Each color is repre-



$\mathcal{N}(\mathbf{0},(\sigma_\epsilon^2+\sigma_{\ell,\ell'}^2)\mathbf{I}_m)$
$\mathbf{U}_{\ell,\ell'}$
$\mathbf{X}_{s_\ell}$
$\mathbf{X}_{s_{\ell'}}$
$=$
$+$
$=$
$\mathbf{U}_{\ell,\tilde{\ell}}$
$\mathbf{U}_{\ell',\tilde{\ell}'}$
$+$
$+$
$\mathcal{N}(\mathbf{0},(\sigma_\epsilon^2+\sigma_{\ell,\tilde{\ell}}^2)\mathbf{I}_m)$
$\mathcal{N}(\mathbf{0},(\sigma_\epsilon^2+\sigma_{\ell',\tilde{\ell}'}^2)\mathbf{I}_m)$
$\mathbf{X}_{s_{\tilde{\ell}}}$
$\mathbf{X}_{s_{\tilde{\ell}'}}$
$=$
$+$
$=$
$\mathbf{U}_{\tilde{\ell},\tilde{\ell}'}$
$\mathcal{N}(\mathbf{0},(\sigma_\epsilon^2+\sigma_{\tilde{\ell},\tilde{\ell}'}^2)\mathbf{I}_m)$

Fig. 1. Factor graph representation of the prior model with $\{(\ell,\ell'),(\ell,\tilde{\ell}),(\ell',\tilde{\ell}'),(\tilde{\ell},\tilde{\ell}')\}\subset B$.

sented by a vector (L^*,a^*,b^*), where L^* is the measure of lightness, a^* represents color from cyan to magenta, and b^* represents color from blue to yellow.

3. EDGE DETECTION

The segmentation algorithm of this paper iterates between two steps. Step 1 is a new edge detection algorithm and Step 2 removes (cuts) the detected edges. By "edge", we mean here an element (ℓ,ℓ') of B that corresponds to an edge in the image. In the end, when the algorithm terminates successfully, such ℓ and ℓ' belong to different segments.

Step 2 of the algorithm is computationally trivial and will be discussed in Section 4. We now focus on Step 1.

3.1. Statistical Model

The description of Step 1 begins by formulating an estimation problem, where the given image $\mathbf{y}=(\mathbf{y}_{s_1},\ldots,\mathbf{y}_{s_L})^\mathsf{T}$ (with $\mathbf{y}_{s_\ell}\in\mathbb{R}^m$) is viewed as a noisy observation of a hypothetical "true" image $\mathbf{X}=(\mathbf{X}_{s_1},\ldots,\mathbf{X}_{s_L})^\mathsf{T}$ (with $\mathbf{X}_{s_\ell}\in\mathbb{R}^m$) with

$$p(\mathbf{y}|\mathbf{x})=\prod_{\ell=1}^{L}\frac{1}{(2\pi\sigma_Z^2)^{\frac{m}{2}}}\exp\left(-\frac{\|\mathbf{y}_{s_\ell}-\mathbf{x}_{s_\ell}\|^2}{2\sigma_Z^2}\right),\quad (1)$$

where the noise σ_Z^2 is a parameter of the algorithm.

The interesting part of the statistical model is the (improper) prior, which penalizes the differences between neighboring pixels as follows. For each pair $(\ell,\ell')\in B$, we define the slack variable

$$\mathbf{U}_{\ell,\ell'}=\mathbf{X}_{s_\ell}-\mathbf{X}_{s_{\ell'}}\quad (2)$$

with a joint probability law $p(\mathbf{u}|\boldsymbol{\sigma}^2)$ (with parameters $\boldsymbol{\sigma}^2$) described below (see also Fig. 1). Writing (2) as $\mathbf{U}=D\mathbf{X}$ with a pertinent matrix D, we define our (improper) prior as

$$\tilde{p}(\mathbf{x};\boldsymbol{\sigma}^2)=\int\delta(\mathbf{u}-D\mathbf{x})\,p(\mathbf{u}|\boldsymbol{\sigma}^2)\,\mathrm{d}\mathbf{u}.\quad (3)$$

Concerning $p(\mathbf{u}|\boldsymbol{\sigma}^2)$, we define $\mathbf{U}_{\ell,\ell'}$ to be independent zero-mean Gaussian random vectors with covariance matrices $\mathbf{I}_m(\sigma_\epsilon^2+\sigma_{\ell,\ell'}^2)$, where σ_ϵ^2 is a parameter of the algorithm and $\sigma_{\ell,\ell'}^2$ are unknown and will be estimated. Our statistical model is thus finalized by

$$p(\mathbf{u}|\boldsymbol{\sigma}^2)=\prod_{(\ell,\ell')\in B}\frac{1}{(2\pi(\sigma_\epsilon^2+\sigma_{\ell,\ell'}^2))^{\frac{m}{2}}}\exp\left(-\frac{\|\mathbf{u}_{\ell,\ell'}\|^2}{2(\sigma_\epsilon^2+\sigma_{\ell,\ell'}^2)}\right),\ (4)$$

where $\boldsymbol{\sigma}^2$ is the vector of all $\sigma_{\ell,\ell'}^2$ with $(\ell,\ell')\in B$. The structure of this prior is illustrated in Fig. 1.

This prior is sparsifying in the following sense: at any local maximum of the likelihood (6), many estimated variance parameters $\sigma_{\ell,\ell'}^2$ are likely to be zero. The corresponding pixel differences (2) are then regularized to be small (but not necessarily zero), thus prompting smooth areas in the image. On the other hand, when the estimated $\sigma_{\ell,\ell'}^2$ is nonzero, the cost for arbitrarily large jumps is very small, which allows for sharp edges in the image [10, 12, 13]. The sparsity level can be adjusted by tuning σ_ϵ^2.

We now detect edges in the image simply as follows: if the estimated $\sigma_{\ell,\ell'}^2$ is nonzero, we decide that there is an edge between s_ℓ and $s_{\ell'}$; otherwise, we decide that there is no edge.

3.2. Maximum "Likelihood" Estimation and EM

As mentioned, we wish to estimate the unknown variances $\boldsymbol{\sigma}^2$ by maximizing the "likelihood"

$$\tilde{p}(\mathbf{y};\boldsymbol{\sigma}^2)=\int p(\mathbf{y}|\mathbf{x})\tilde{p}(\mathbf{x};\boldsymbol{\sigma}^2)\,\mathrm{d}\mathbf{x}\quad (5)$$

$$=\int\int p(\mathbf{y}|\mathbf{x})\delta(\mathbf{u}-D\mathbf{x})p(\mathbf{u}|\boldsymbol{\sigma}^2)\,\mathrm{d}\mathbf{u}\,\mathrm{d}\mathbf{x}.\quad (6)$$

To this end, we use expectation maximization (EM) as follows. Considering $\mathbf{U}$ as a hidden variable, we update $\boldsymbol{\sigma}^2$ iteratively by

$$\hat{\boldsymbol{\sigma}}_{k+1}^2=\underset{\boldsymbol{\sigma}^2}{\operatorname{argmax}}\,\mathbb{E}\big[\ln\tilde{p}(\mathbf{y},\mathbf{U};\boldsymbol{\sigma}^2)\big]\,,\quad (7)$$

with iteration index k and starting from an initial guess $\hat{\boldsymbol{\sigma}}_0^2$. The expectation in (7) is taken with respect to the density $p(\mathbf{u}|\mathbf{y},\hat{\boldsymbol{\sigma}}_k^2)$. It can be seen from (6) that the maximization problem splits for each $\sigma_{\ell,\ell'}^2$:

$$\hat{\sigma}_{\ell,\ell'}^2=\underset{\sigma_{\ell,\ell'}^2}{\operatorname{argmax}}\,\mathbb{E}\big[\ln p(\mathbf{U}_{\ell,\ell'}|\sigma_{\ell,\ell'}^2)\big]\quad (8)$$

$$=\max\left(0,\frac{1}{m}\,\mathbb{E}\Big[\|\mathbf{U}_{\ell,\ell'}\|^2\Big]-\sigma_\epsilon^2\right),\quad (9)$$

3.3. Scalarization

For grayscale images (i.e. $m=1$), $p(\mathbf{u},\mathbf{x},\mathbf{y}|\boldsymbol{\sigma}^2)$ can be represented by a factor graph as in Fig. 2. For color images

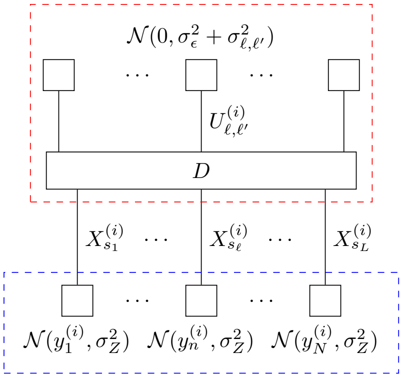

$\mathcal{N}(0, \sigma_\epsilon^2 + \sigma_{\ell,\ell'}^2)$
$\cdots$
$\cdots$
$U_{\ell,\ell'}^{(i)}$
D
$X_{s_1}^{(i)}$
$\cdots$
$X_{s_\ell}^{(i)}$
$\cdots$
$X_{s_L}^{(i)}$
$\cdots$
$\cdots$
$\mathcal{N}(y_1^{(i)}, \sigma_Z^2)$
$\mathcal{N}(y_n^{(i)}, \sigma_Z^2)$
$\mathcal{N}(y_N^{(i)}, \sigma_Z^2)$

Fig. 2. Factor graph representation of $p(\mathbf{u}, \mathbf{x}, \mathbf{y}|\boldsymbol{\sigma}^2)$, for each component $i \in \{1, 2, 3\}$ of the color image model. Top dashed box: $\tilde{p}(\mathbf{x}^{(i)}; \boldsymbol{\sigma}^2)$. Bottom dashed box: $p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$.

($m = 3$), we first note that

$$\mathbb{E}\Big[\|\mathbf{U}_{\ell,\ell'}\|^2\Big] = \sum_{i=1}^{3} \mathbb{E}\Big[(U_{\ell,\ell'}^{(i)})^2\Big]. \tag{10}$$

Moreover, it is easily seen that, the factor graph of $p(\mathbf{u}, \mathbf{x}, \mathbf{y}|\boldsymbol{\sigma}^2)$ also splits into three independent factor graphs, one for each color channel, each of which with the structure of Fig. 2.

3.4. Iterative Scalar Gaussian Message Passing

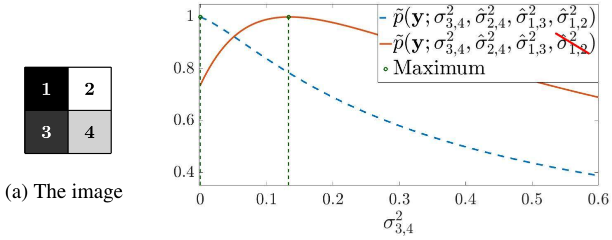
For complexity reasons, we do not attempt to compute $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$ exactly. Instead, we use iterative scalar Gaussian message passing [10, 15] in the factor graph(s) of Fig. 2 to compute an approximate estimate of $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$, which turns out to work well in practice. If the algorithm converges, the estimated means are correct, but the variances are not [17, 19]. The details of this computation can be obtained through specializing the algorithm of [10] by setting the matrix A to an identity matrix. Note also that the computational complexity scales linearly with the number of pixels. After the algorithm has converged, $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$ can be calculated from the messages in the factor graph, and $\hat{\sigma}_{\ell,\ell'}^2$ can then be obtained from (9).

4. ITERATIVE EDGE CUTTING

4.1. The Algorithm

The edge detection algorithm of Section 3 is executed repeatedly. After each execution, the detected edges (i.e., the pairs $(\ell, \ell') \in B$ with $\sigma_{\ell,\ell'}^2 > 0$) are added to the set $E \subset B$ of detected edges and cut. Cutting an edge means fixing $\sigma_{\ell,\ell'}^2 = \infty$ or, equivalently, removing the corresponding column of D for all subsequent iterations. The iteration is continued until no new edges are detected.

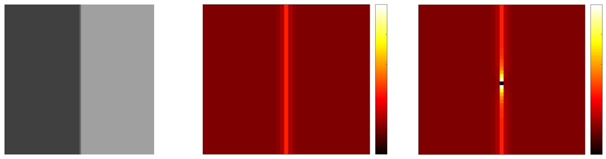
Somewhat surprisingly, this algorithm empirically works well and yields satisfactory closed segments. We illustrate the effects of edge cutting by three examples.



(a) The image

(b) "Likelihood" function

Fig. 3. Edge cutting in a 4-pixel image: the "likelihood" (as a function of $\sigma_{3,4}^2$) before and after cutting the edge $(1, 2)$.



(a) The image

(b) Heat map before cut

(c) Heat map after cut

Fig. 4. A simple two-color image and the calculated value of $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$, represented by a heat map, before and after cutting the horizontal edge in the center of the image.

4.2. Toy Example 1

Consider the example in Fig. 3 (a), which is a grayscale image with only four pixels $\{1, 2, 3, 4\}$. The blue dashed curve in Fig. 3 (b) shows the "likelihood" $\tilde{p}(\mathbf{y}; \boldsymbol{\sigma}^2)$ as a function of $\sigma_{3,4}^2$ while the other $\sigma_{\ell,\ell'}^2$ are fixed to their optimum. Note that this curve reaches its maximum for $\sigma_{3,4}^2 = 0$. After cutting the edge $(1, 2)$, the "likelihood" function changes into the red solid curve in Fig. 3 (b), which reaches its maximum for $\sigma_{3,4}^2 > 0$. In other words, cutting the edge $(1, 2)$ entails the subsequent detection of an edge $(3, 4)$.

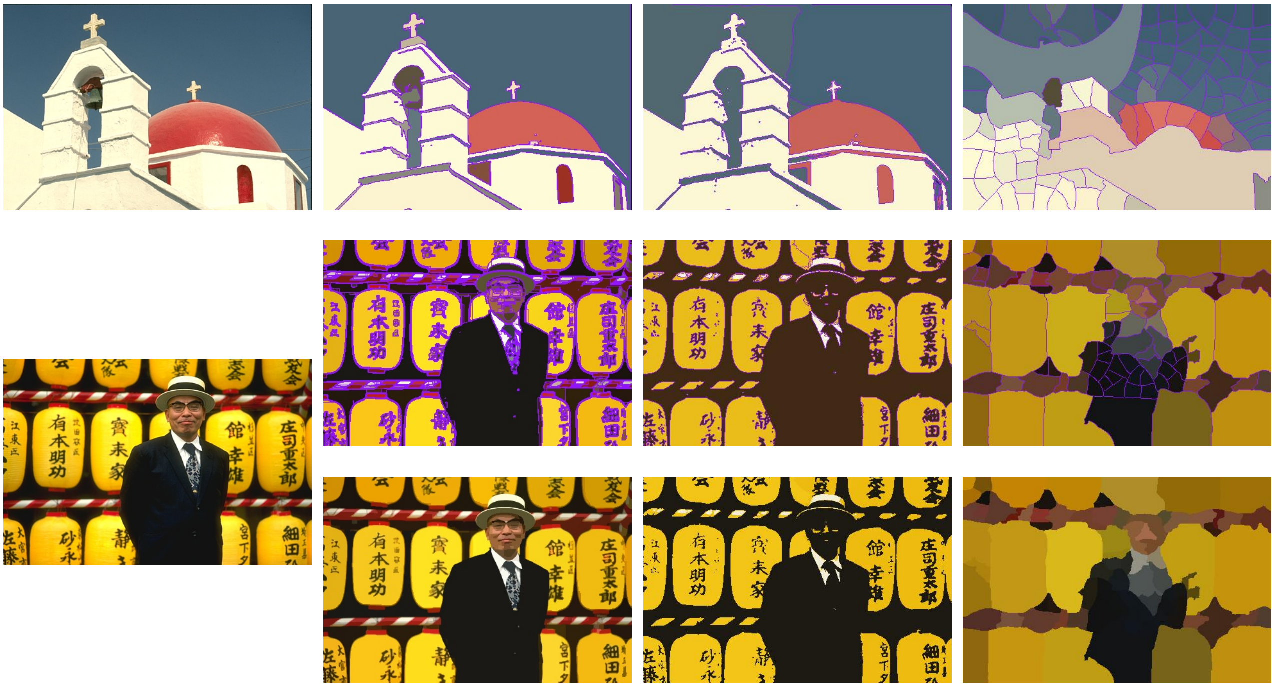
4.3. Toy Example 2

In our second example, we focus on the quantity $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$. It is obvious from (9) that a sufficiently large value of $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$ leads to an edge. The example is shown in Fig. 4. The middle picture shows the approximate value of $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$ (as computed by iterative scalar Gaussian message passing) for horizontally neighboring pixels as a heat map, i.e., with brighter colors indicating higher values.

We then cut a single horizontal edge in the center of the image. The effect of this cut on $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$ is shown in the rightmost picture in Fig. 4. Note that $\mathbb{E}\big[\|\mathbf{U}_{\ell,\ell'}\|^2\big]$ now becomes larger for horizontally neighboring pixels close to the cut edge, which leads to edge detections near this cut.

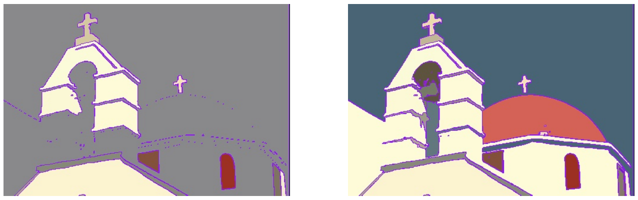
4.4. Example 3

The progress of iterative edge cutting is also demonstrated in Fig. 6, which is discussed in the next section.



(a) Original images (b) Proposed algorithm (c) Mean shift [2] (d) Normalized cuts [7]

Fig. 5. Experimental results from the proposed method, mean shift clustering [2] and the normalized cuts method [7]. First and second row: boundaries of segments are marked with purple lines. Third row: segment boundaries are not marked.



(a) After the first EM update (b) After the 20th EM update

Fig. 6. Edges and segments after the first and after the 20th iteration of the proposed algorithm.

5. EXPERIMENTAL RESULTS

Fig. 5 and Fig. 6 show some experimental results using two test images from the Berkeley Segmentation Dataset (BSDS300) [20]. The parameters of our algorithm were set to $\sigma_\epsilon^2 = 10^{-4}$ and $\sigma_Z^2 = 8 \cdot 10^{-4}$ (same for both images). For the edge detection (Section 3), we used 20 iterations of Gaussian message passing followed by a single EM update. In the outer loop (Section 4.1), the edge detection was executed 20 times. In these examples, our algorithm took about 2 seconds on an ordinary laptop computer.

For comparison, Fig. 5 also shows the results of two other segmentation methods: mean shift [2] and normalized cuts [7]. We also experimented with two other standard algorithms, K-means [21] and the watershed transformation [22], the results of which are inferior to all the methods in Fig. 5. In Fig. 5 and 6, all pixels within the same segment are given the same color: the average of all the corresponding pixels in the original image. In the first two rows of Fig. 5, the detected edges ($\approx$ the segment boundaries) are marked in purple.

From Fig. 5 and many other examples (not reported here), we conclude that the proposed algorithm is not inferior to, and arguably better than, the mean shift algorithm, and clearly outperforms the other methods. This conclusion is supported also by numerical scores using the (different) metrics defined in [23, 24]. In any case, our approach does not wash out small details if they sufficiently contrast with their background. Concerning the mean shift algorithm, note the crack in the sky behind the church, and the merging of the red parts of the horizontal ribbons with the background in the second image.

Fig. 6 shows how the segmentation result evolves during the algorithm. After the first pass of the edge detection algorithm, many edges are still isolated and there are only few closed segments. After 20 iterations, we observe the reverse: most edges now belong to clear segment boundaries.

6. CONCLUSION

We proposed a new approach for image segmentation based on a graphical model with NUV terms, which favors piecewise smooth segments. The proposed algorithm iterates two steps. The first step uses approximate EM for edge detection, with computations amounting to iterative scalar Gaussian message passing. In the second step, the edges detected in the first step are removed (cut). In our simulations, the proposed method yields closed segments and converges quite fast, and the results compare favorably to four other methods.

7. REFERENCES

[1] K. K. Singh and A. Singh, "A study of image segmentation algorithms for different types of images," *International Journal of Computer Science*, vol. 7, no. 5, pp. 414–417, 2010.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[3] N. Senthilkumaran and R. Rajesh, "Edge detection techniques for image segmentation–a survey of soft computing approaches," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp. 250–254, 2009.

[4] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, "Recent advances in graph partitioning," in *Algorithm Engineering*, pp. 117–158. Springer, 2016.

[5] R. F. Moghaddam and M. Cheriet, "Adotsu: An adaptive and parameterless generalization of Otsu's method for document image binarization," *Pattern Recognition*, vol. 45, no. 6, pp. 2419–2431, 2012.

[6] K. Parvati, P. Rao, and M. Mariya Das, "Image segmentation using gray-scale morphology and marker-controlled watershed transformation," *Discrete Dynamics in Nature and Society*, 2008.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[8] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1382–1389, 2007.

[9] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, 1993.

[10] N. Zalmai, C. Luneau, C. Stritt, and H.-A. Loeliger, "Tomographic reconstruction using a new voxel-domain prior and Gaussian message passing," in *Proc. 24th European Signal Processing Conf. (EUSIPCO)*, 2016, pp. 2295–2299.

[11] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[12] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[13] H.-A. Loeliger, L. Bruderer, H. Malmberg, F. Wadehn, and N. Zalmai, "On sparsity by NUV-EM, Gaussian message passing, and Kalman smoothing," *Information Theory and Applications Workshop (ITA)*, Feb 2016.

[14] S. Som and P. Schniter, "Approximate message passing for recovery of sparse signals with Markov-random-field support structure," *International Conf. on Machine Learning*, 2011.

[15] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.

[16] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, Li Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[17] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, 2001.

[18] J. Schanda, *Colorimetry: Understanding the CIE System*, John Wiley & Sons, 2007.

[19] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *NIPS*, 2000, vol. 13, pp. 689–695.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th International Conf. Computer Vision*, July 2001, vol. 2, pp. 416–423.

[21] M. C. Chandhok, S. Chaturvedi, and A. Khurshid, "An approach to image segmentation using K-means clustering algorithm," *International Journal of Information Technology*, vol. 1, no. 1, pp. 11–17, 2012.

[22] A. Bala, "An improved watershed image segmentation technique using MATLAB," *International Journal of Scientific & Engineering Research*, vol. 3, no. 6, pp. 1–4, 2012.

[23] J. Pont-Tuset and F. Marques, "Measures and meta-measures for the supervised evaluation of image segmentation," in *Computer Vision and Pattern Recognition*, 2013.

[24] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 530–549, 2004.