# EFFICIENT PROPOSAL DISTRIBUTIONS FOR MCMC IMAGE SEGMENTATION

*Timo Kostiainen, Jouko Lampinen*

Helsinki University of Technology
Laboratory of Computational Engineering
P.O.Box 9203, FIN-02015 FINLAND
Timo.Kostiainen@hut.fi, Jouko.Lampinen@hut.fi

## ABSTRACT

We present methods to obtain computationally efficient proposal distributions for Bayesian reversible jump Markov chain Monte Carlo (RJMCMC) based image segmentation. The slow convergence of MCMC methods often makes them poorly suited for practical image processing applications. We show how carefully crafted proposal distributions along with certain approximations can decrease the computational cost of MCMC image segmentation to a level that is comparable with some traditional algorithms. We also discuss the interpretation of the resulting distribution of different segmentations and present experimental results.

## 1. INTRODUCTION

Bayesian methods of computer vision are based on defining a posterior probability distribution for unknown parameters of the problem and attempting to recover that distribution using the observed image data. Computer vision problems can rarely be solved in closed form and the use of Markov chain Monte Carlo (MCMC) techniques is a common solution. Image segmentation based on Bayesian MCMC methods has been studied before, usually using Markov random fields [1]. Typically estimation of the distribution is done in a purely top-down manner: random samples are picked and their fit to the data is evaluated. In high dimensional problems, a very large number of samples are required and convergence takes a long time.

Clark and Quinn [2] have presented an MCMC image segmentation algorithm where data driven proposal samples are obtained using image information and used to increase the acceptance rates in the sampling chain. Image segments are modeled by Gaussian intensity distributions, with a Markov random field model for prior probabilities of segment labels. Tu *et al.* [3] have proposed another data driven MCMC approach to image segmentation, in which various types of cues from image data are used to drive the MCMC algorithm, that is, to produce good proposition samples using bottom-up information processing. The probability model for the observed image is based on different texture models for the image segments. In this paper we apply the same principles to construct a general purpose image segmentation method for use as a basis for further image processing. The goal is a rough segmentation of different objects into different segments. Our emphasis is on minimizing computational complexity without allowing any significant degradation in the quality of results.

## 2. PROBABILITY MODEL

The segment division is defined by assigning a segment label to each image pixel. The segments have continuous boundaries. A different texture model, controlled by parameters $\theta_s$, is associated with each image segment. We treat the segments as independent of each other, so the likelihood of the image $I$ with segmentation $\mathbf{S}$ and segment model parameters $\theta$ is the product of segment likelihoods

$$p(I|\mathbf{S},\theta,M) = \prod_s p(I_s|\mathbf{S},\theta_s,M), \tag{1}$$

where $I_s$ is the part of the image that belongs to segment $s$ and $M$ represents the constraints included in the model.

The posterior distribution of the segment division $\mathbf{S}$ and model parameters $\theta$, given the observed image $I$, is computed as follows:

$$p(\mathbf{S},\theta|I,M) = \frac{p(I|\mathbf{S},\theta,M)p(\mathbf{S},\theta|M)}{p(I|M)}. \tag{2}$$

We are interested in the segment division of the image, not the texture models. Therefore, we choose the prior distribution of the segmentation state $p(\mathbf{S}|M)$ to be independent of $\theta$ and integrate over $\theta$ to obtain the posterior distribution of the segment division:

$$p(\mathbf{S}|I,M) = \frac{\int p(I|\mathbf{S},\theta,M)p(\theta|M)d\theta p(\mathbf{S}|M)}{p(I|M)}. \tag{3}$$

Computation of the prior is discussed in section 2.2. We estimate the posterior distribution (3) using MCMC sampling.

0-7803-8554-3/04/$20.00 

For computational efficiency, we approximate the integral by replacing the distribution of $\theta$ with a delta function centered at the maximum of the conditional posterior distribution $p(\theta|I, S, M)$. This approximation leads to an empirical Bayes method and theoretically, it has the effect of over-fitting the rest of the parameters, namely $\mathbf{S}$, but the following arguments support our choice:

- We choose simple texture models which have unimodal likelihood functions
- The texture models have only a few degrees of freedom, and a validation technique is used to prevent over-fitting
- The likelihood functions are expected to be highly peaked, and experimental results indicate that the difference between the likelihood of the point estimate and the expected likelihood obtained by taking samples from the conditional posterior is negligible
- Savings in computational cost are substantial
- Image segmentation lacks a theoretical foundation, which is why the probability model in itself is a gross simplification.

### 2.1. Texture models

The likelihood computation is based on texture models $\theta$. Our goal is not to produce an accurate model that is able to generate a realistic-looking reconstruction of the image in terms of the segment division and the texture models. Instead, we seek a rough segmentation where different-looking objects should be classified into different segments. For this purpose we find that the texture models for the segments should have low complexity. It is worth noting that there is no theoretically correct way to define the general segmentation of natural images in terms of texture models, so any choice is a heuristic approximation, and its goodness can only be measured by subjective evaluation of practical results.

We choose segment texture models using partial predictive likelihoods of the texture models, that is, divide pixels into training and test sets. We evaluate the texture models in the *YCbCr* color space. The spectral channels are modeled independently, using one of four different probability distribution models for textures: Gaussian, Laplacian, multinomial, or – for the luminance channel only – a linear spatial model with additive Gaussian noise. In all these models, pixel probabilities are assumed to be independent and the likelihood of the segment is the product of pixel likelihoods. The multinomial model has a high number of degrees of freedom, and to avoid overfitting we apply a Dirichlet distibution as its conjugate prior. For the rest of the models, which have no more than a few free parameters, we use non-informative uniform priors.

### 2.2. Prior distributions

We use quite general prior distributions to control properties of the segments. They are not tuned for any particular type of scene. We use a smoothness prior for segment outlines, a Poisson prior for the number of segments and a $1/x$ shaped prior for segment sizes.

## 3. MCMC PROCESS

We use a reversible jump MCMC technique [4] to obtain samples of the posterior probability distribution. The samples are different segment divisions of the image. Proposal samples are generated and either accepted or rejected with a probability that maintains the balance of the MCMC chain. Ehlers and Brooks [5] have shown that if in the RJMCMC jump from dimension $k$ to $k'$, the variables specific to $k'$ are generated from their conditional posterior distributions, conditioned on the variables common to $k$ and $k'$, then the generated variables do not contribute to the acceptance probability of the jump. We use this result and generate the parameters of the texture models for segments created in the MCMC transitions by approximating their posterior distributions.

### 3.1. MCMC state transitions

Different data driven techniques are applied in the generation of the proposal samples. We use three different transition processes: diffusion, split and merge. Diffusion does not affect dimension of the parameter space, while split and merge form a pair of opposite transitions between different dimensions. Together, the processes allow reversible jumps between the segmentation states.

The diffusion routine draws the proposition samples directly from the likelihood. We randomly select one of the segments, $s_g$, for region growing. All segments have an equal probability of being selected. We choose a random radius value $r$ for a circular diffusion kernel. We dilate the selected segment by moving the kernel along its boundary and obtain $s'_g$. For each pixel on the diffusion area, , $x_i|(x_i \in s'_g, x_i \notin s_g)$, we compute two likelihood values: the likelihood $p(x_i|\theta_o)$ for the model $\theta_o$ of the segment that the $x_i$ currently belongs to and $p(x_i|\theta_g)$ for the model $\theta_g$ of the growing segment $s_g$. The likelihood ratio $\gamma(x_i) = p(x_i|\theta_g)/p(x_i|\theta_o)$ is used to determine which pixels should be transferred to segment $s_g$. We use a spatial low-pass filter with a radius proportional to $r$ to smooth the values of $\gamma(x_i)$ between neighboring pixels in order to help keep the segments contiguous. The filtered values of the likelihood ratio determine which pixels are proposed to

be transferred. It is possible that the balance of the MCMC chain is not strictly preserved in the diffusion step. Experimentally, we have not encountered any problems due to this.

Split attempts to divide a segment into two new segments along apparent edges (found by filtering). First, one of the segments is selected for splitting, and a starting point is picked within that segment. These choices are based on using the likelihood values of segments and pixels, such that badly fitting areas and segments are likely to be targeted. Finally, an edge tracing method is applied to produce a division of the segment. Acceptance probabilities are computed based on the distributions from which the samples are generated and the probabilities of opposite transitions.

Merge proposes joining two randomly selected adjacent segments. Kullback-Leibler distances between segment probability models are used in the generation process.

### 3.2. Analysis of the posterior distribution

The result of MCMC sampling is a set of samples from the posterior distribution. In the context of image segmentation, it is not obvious how to combine the different segmentations in a way that is easy to interpret. The result is a set of $N$ sample segmentations $\mathcal{S} = \{\mathbf{S}^1, \ldots, \mathbf{S}^N\}$, where each $\mathbf{S}^i = \{S_1^i, \ldots, S_{M_i}^i\}$ consists of $M_i$ segments. To compute an average of these, we suggest the following scheme, which is based on the co-occurrence of pixels in the same segment: 1) Choose a pivot pixel $x_p(s)$ for each segment label $s$. $x_p(s)$ is any of those pixels that most often get classified to the segment $s$. Discard those segment labels that get winning support in no pixel. 2) For each pixel $x$, compute the number of samples in which the pixel gets the same label as pivot pixel $x_p(s)$, that is, $\eta(x, x_p(s)) = |\{\mathbf{S}^k \subset \mathcal{S} : \{S_j^k \in \mathbf{S}^k : x, x_p(s) \in S_j^k\}\}|$, where $|\cdot|$ denotes set cardinality. 3) The expected label $\hat{s}(x)$ of the pixel $x$ is the value of $s$ that maximizes $\eta(x, x_p(s))$.

The value $c(x) = \max_s\{\eta(x, x_p(s))\}/\sum_s \eta(x, x_p(s))$ can be used as a measure of confidence in the classification of pixel $x$.

## 4. RESULTS

The comparison of different segmentation algorithms is difficult because there is no single ground truth. In order to obtain results similar to those of human subjects, the computer would have to recognize the objects. Recognition of objects in the scene cannot be simulated by a simple cost function. The results will also depend on the desired detail level. For example, a face could be one segment, or the eyes, the mouth and the nose might all be separate segments. Even high-level recognition and interpretation of the image does not resolve this ambiguity.

We have tested our method on a large database of natural images, and we find that the results are on a par with other published segmentation techniques. Examples are shown in Fig. 1, and more are available at our web site[1].

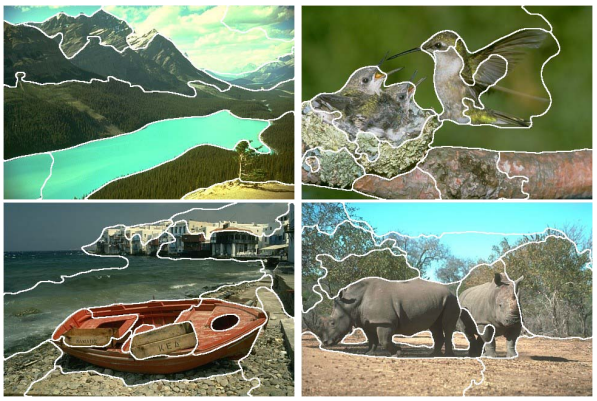

**Fig. 1**. Examples of segmentation results.

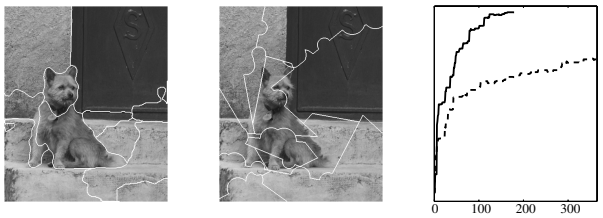


**Fig. 2**. Data driven vs. random proposals. *Left*: result of data driven segmentation after 179 samples. *Center*: result of segmentation using random proposal distributions in the same amount of CPU time (365 samples). Metropolis-Hastings acceptance probabilities were 27 % and 20 %, respectively. *Right*: evolution of log posterior probabilities $\log p(\mathbf{S}|I, M)$ in the MCMC chains with data driven (*continuous*) and random (*dashed*) proposal distributions.

Due to efficient generation of proposal samples, especially the diffusion algorithm, most of the computation time is spent evaluating the segment likelihoods. With $300 \times 400$ pixel images, running Matlab on a 1.8 GHz processor, 5 – 10 MCMC samples per second can be processed. Upward of one hundred samples are required for practically sufficient convergence, so running times per image range between 20 – 60 seconds. Tu and Zhu report processing times of 10 – 50 minutes for their data driven MCMC algorithm in a similar setting [3].

In order to evaluate the advantage of the data driven proposal distributions, in Fig. 2 we compare the performance of the proposed algorithm to that of a sampling chain in which the proposal samples are random and do not depend on the data.

[1] http://www.lce.hut.fi/research/compinf/segment/

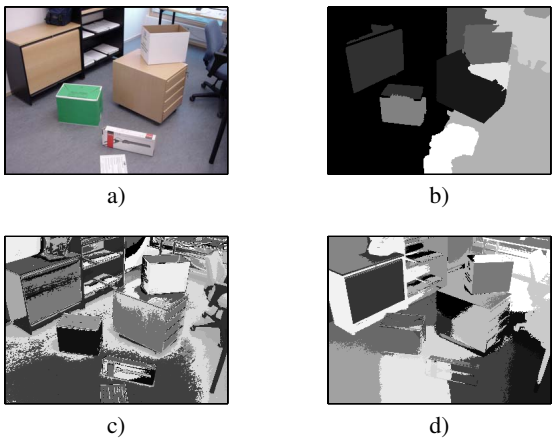

**Fig. 3**. Comparison with segmentation methods based on clustering. Segments are displayed using different grayscales in no particular order. *a)* Original target image. *b)* Segmentation produced by the MCMC based method with a running time equal to that in *d)*. *c)* Results of the FCM method [6] *d)* Fuzzy c-means clustering of color channels and spatial coordinates.

In Fig. 3 we compare our method to two segmentation algorithms based on color clustering. A simple segmentation method to produce spatially confined segments can be obtained by including image coordinates as input variables to the clustering algorithm in addition to the three color channels. We applied the fuzzy c-means algorithm directly to do the segmentation on the five dimensional input data. We scaled the variance of the coordinate values to twice the variance of the RGB channels. The result appears in Fig. 3*d*. This algorithm has a convergence time that is approximately equal to that of our MCMC method for 0.1 – 1 megapixel image sizes. The result that the MCMC method gives in an equal CPU time is shown in Fig. 3*b*. Both were implemented in Matlab. The algorithm presented in ref. [6] (Fig. 3*c*) analyses color channel histograms to determine thresholds for segment classification, and applies fuzzy c-means clustering to produce a vector quantization in color space. We applied a C implementation which runs roughly twice as fast as the other two algorithms.

The example shows that MCMC techniques, despite their reputation as being frustratingly slow, can be quite competitive against classical heuristic techniques if they are designed to combine different types of information effectively. The use of bottom-up information in the generation of MCMC samples yields substantial savings in computation time.

## 5. CONCLUSION

We have described a Bayesian MCMC based image segmentation method that produces results that compare favorably with some simpler segmentation algorithms whose computational costs are of the same order of magnitude. Advantages of a model-based, probabilistic approach are numerous. The modular structure allows the inclusion of different texture models and new methods for producing proposal samples for the MCMC algorithm. The model provides a framework for scene analysis, since it gives a probabilistic explanation to each part of the image. Prior probabilities for any properties of the segments can be used directly to control the results. The quality of the data-driven proposal samples is critical to performance.

The quality of results is limited by the simplicity of the image model – independent segments that follow statistical texture distributions. However, we expect that benefits of using more complex and more realistic texture models would not be dramatic. A severe problem with many richer texture models is their sensitivity to high spatial frequencies which can typically be found at the boundaries of segments, not within them. A semantically correct hierarchical segmentation necessarily requires recognition of objects and understanding of the 3-D structure of the scene. The approach of modeling the image as patches of different statistically defined textures obviously cannot accomplish this.

## 6. REFERENCES


[1] S. A. Barker and P. J. W. Rayner, "Unsupervised image segmentation using Markov random field models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 1997, pp. 165–178.

[2] E. Clark and A. Quinn, "A data-driven Bayesian sampling scheme for unsupervised image segmentation," in *International Conference on Acoustic, Speech and Signal Processing*, 1999.

[3] Tu Zhuowen and Song-Chun Zhu, "Image segmentation by data-driven Markov chain Monte Carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657–673, 2002.

[4] P. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrica*, vol. 82, no. 4, pp. 711–732, 1995.

[5] R.S Ehlers and S.P. Brooks, "Constructing general efficient proposals for reversible jump MCMC," Available online at http://www.est.ufpr.br/rt/ehlb03.htm, July 2003.

[6] Young Won Lim and Sang Uk Lee, "On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 935–952, 1990.