Unsupervised Image Segmentation using Markov Random Field Models

S.A.Barker and P.J.W.Rayner

Signal Processing and Communications Group,
Cambridge University Engineering Dept.,
Cambridge CB2 1PZ, England

Abstract. We present an unsupervised segmentation algorithm based on a Markov Random Field model for noisy images. The algorithm finds the the most likely number of classes, their associated model parameters and generates a corresponding segmentation of the image into these classes. This is achieved according to the MAP criterion. To facilitate this, an MCMC algorithm is formulated to allow the direct sampling of all the above parameters from the posterior distribution of the image. To allow the number of classes to be sampled, a reversible jump is incorporated into the Markov Chain. The jump enables the possible splitting and combining of classes and consequently, their associated regions within the image. Experimental results are presented showing rapid convergence of the algorithm to accurate solutions.

1 Introduction

The segmentation of noisy or textured images into a number of different regions comprises a difficult optimisation problem. This is compounded when the number of regions into which the image is to be segmented is also unknown.

If each region within an image is described by a different model class then the observed image may be viewed as a sample from a realisation of an underlying class map. Image segmentation can therefore be treated as an incomplete data problem, in which the intensity data is observed, the class map is missing and the associated class model parameters need to be estimated. In the unsupervised case, the number of model classes may also be unknown.

The unsupervised segmentation problem has been approached by several authors. [1] [7] and [8] propose algorithms comprising of two steps. The image is assumed to be composed of an unknown number of regions, each modelled as individual Markov Random Fields.

The first of these steps is a coarse segmentation of the image into the most 'likely' number of regions. This is achieved by dividing the image into windows, calculating features or estimating model parameters, then using a measure to combine closely related windows. The resulting segmentation is then used to estimate model parameters for each of the classes, before a supervised high-resolution segmentation is carried out via some form of relaxation algorithm.

A similar methodology is used in [5] but the measure used, the Kolmogorov-Smirnov distance, is a direct measure of similarity of the distributions of grayscale values (in the form of histograms) between adjacent windows. Windows are then combined into a single region if the distance between their distributions is relatively small. A variant of this algorithm [2] is based on the same distance function but the distribution of grayscales in each window is compared with the distribution functions of the samples comprising each class over the complete image. If the distribution of one class is found to be close enough to that of the window then it is designated as being a member of that class. Otherwise, a new outlier class is created. When the field stabilises, usually after several iterations, new classes are created from the outliers if they constitute more than one percent of the image. If not, the algorithm is re-run.

A split and merge algorithm is proposed in [6]. The image is initially split into large square windows but these are then re-split to form four smaller windows if a uniformity test for each window is not met. The process ends when windows as small as 4 × 4 pixels are reached. The windows are then merged to form regions using a distance measure based on the pseudo-likelihood.

In [3] segmentations and parameter estimates were obtained by alternately sampling the label field and calculating maximum pseudo likelihood estimates of the parameter values. The process was repeated over differing numbers of label classes and the resulting estimates were applied to a model fitting criteria to select the optimum number of classes and hence, the image segmentation. The criterion used compensates the likelihood of the optimised model with a penalty term that offsets image size against the number of independent model parameters used. The penalty term and its associated parameter values were selected arbitrarily.

This method of exhaustive search over a varying number of classes was developed further in [4]. An EM algorithm was first used to estimate parameters while alternately segmenting the image. To select between the resulting optimsations of the differing models the function of increasing likelihood against increasing model order was fitted to a rising exponential model. The exponential model parameters were selected in a least squares sense. The optimum model order was then found at a pre-specified knee point in the exponential curve.

The approach to unsupervised segmentation presented here comprises a Markov Chain Monte Carlo (MCMC) algorithm to sample from the posterior distribution so that simulated annealing may be used to estimate the MAP solution. This methodology is similar to that used in [13] to segment an image using a known MRF model. Here the method is extended so that the MAP estimate is over not just the segmentation of the image into classes but the number of classes and their respective model parameters.

The algorithm differs from those reviewed in that no windowing is required to estimate region model parameters. The algorithm's MCMC methodology removes the necessity for an exhaustive search over a subsection of the parameter space. This ensures an improvement in efficiency over algorithms that require a separate optimisation to be carried out for each model before a model order selection is made.

The remainder of this paper is divided as follows: section 2 specifies the image model used throughout the paper. The posterior distribution for this model is derived. Section 3 describes the algorithms employed to sample from this posterior distribution. In section 3.1 the segmentation process or allocation of class labels to pixel sites is described. Section 3.2 shows how the noise and MRF model parameters may be sampled from their conditional densities. Section 3.3 describes how reversible jumps are incorporated into the Markov Chain to enable the number of classes into which the image is to be segmented to be sampled. Experimental results of such an algorithm are presented in section 4 and the paper is concluded in section 5.

2 Image Model

Let Ω denote an $M \times N$ lattice indexed by (i,j) so that $\Omega = \{(i,j); 1 \leq i \leq M,\ 1 \leq j \leq N\}$. Let $Y = \{Y_s = y_s;\ s \in \Omega\}$ be the observed grayscale image where pixels take values from the interval $(0,1]$. Then let $X = \{X_s = x_s;\ s \in \Omega\}$ correspond to the labels of the underlying Markov Random Field which have values taken from $\Lambda = \{0, 1, ..., k-1\}$.

The neighbours of the pixel at site s are denoted η_s and the model parameter vector is denoted Ψ. If a Gibbs MRF is used to model the image, the joint conditional density for an observed pixel grayscale value and class label at site s may be expressed,

$$p(Y_s = y_s, X_s = x_s \mid \eta_s,\ \Psi) \ = \ \frac{1}{Z_s}\, e^{-U(Y_s = y_s, X_s = x_s \mid \eta_s,\ \Psi)} \tag{1}$$

where $U(\cdot)$ is the energy function and Z_s is the normalising constant or partition function of the Gibbs distribution. If we write $\Psi = [\{\Theta_c,\ \beta_c^{(0)}; c \in \Lambda\},\ \beta^{(1)}]$, where Θ_c corresponds to the noise model parameter vector, $\beta_c^{(0)}$ to the external field parameter, and $\beta^{(1)}$ is the inter-pixel interaction strength, the Gibbs distribution may be factorised so that,

$$\begin{aligned} p(Y_s = y_s, X_s = x_s \mid \eta_s,\ \Psi) \ &= \\ &\frac{1}{Z_s}\, e^{\left\{-U_1(Y_s = y_s \mid X_s = x_s,\ \Theta_{x_s})\ -\ U_2(X_s = x_s \mid \eta_s,\ \beta_{x_s}^{(0)},\ \beta^{(1)})\right\}} \end{aligned} \tag{2}$$

The partition function Z over the complete image is far too complex to evaluate, so it is unfeasible to compare the likelihoods of two differing Markov Random Field realisations. The closest one can come is to compare their Pseudo-Likelihoods [9]. The Pseudo-Likelihood, given the above described model is given,

$$\begin{aligned} PL(Y = y, X = x \mid \Psi) \; = & \\ & \prod_{s \in \Omega} \frac{1}{z_{x_s}(\Theta_{x_s})} \; e^{\{-U_1(Y_s = y_s \mid X_s = x_s, \; \Theta_{x_s})\}} \\ & \times \; \frac{e^{\left\{- \sum_{s \in L} U_2(X_s = x_s \mid \eta_s, \; \beta^{(0)}_{x_s}, \; \beta^{(1)})\right\}}}{\prod_{s \in \Omega} \sum_{c \in \Lambda} e^{\left\{-U_2(X_s = c \mid \eta_s, \; \beta^{(0)}_c, \; \beta^{(1)})\right\}}} \end{aligned} \qquad (3)$$

where $z_c(\Theta_c)$ is the normalising function for the grayscale value conditional distribution. Using Bayes law, the above factorisation of the Pseudo-Likelihood and incorporating the model order k, the posterior probability for the image model may now be approximated,

$$\begin{aligned} p(X = x, \Psi, k \mid Y = y) \; \propto \; & p(Y = y \mid X = x, \Theta) \; PL(X = x \mid \beta) \\ & \times \; \left(\prod_{c=0}^{k-1} p_r(\Theta_c) \; p_r(\beta^{(0)}_c) \right) \\ & \times \; p_r(\beta^{(1)}) \; p_r(k) \end{aligned} \qquad (4)$$

where $p_r(\cdot)$ are priors, β represents the parameter vector $[\beta^{(0)}_c, \beta^{(1)}; c \in \Lambda]$.

Assuming a Gaussian noise model for each of the classes, the parameter vector $\Theta_c = [\mu_c, \sigma_c]$. If a nearest neighbour Gibbs MRF is chosen to model the image, then the posterior density may be written,

$$\begin{aligned} p(X = x, \Psi, k \mid Y = y) \propto & \\ & \frac{1}{\sqrt{2\pi\sigma^2_{x_s}}} \exp\left\{ -\frac{1}{2} \sum_{s \in \Omega} \left(\frac{y_s - \mu_{x_s}}{\sigma_{x_s}} \right)^2 \right\} \\ & \times \; \frac{\exp\left\{ - \sum_{s \in \Omega} \left(\beta^{(0)}_{x_s} + \beta^{(1)} \; V(x_s, \eta_s) \right) \right\}}{\prod_{s \in \Omega} \sum_{c \in \Lambda} \exp\left\{ - \left(\beta^{(0)}_c + \beta^{(1)} \; V(c, \eta_s) \right) \right\}} \\ & \times p_r(\beta^{(1)}) \; p_r(k) \prod_{c \in \Lambda} p_r(\mu_c) \; p_r(\sigma_c) \; p_r(\beta^{(0)}_c) \end{aligned} \qquad (5)$$

where $V(c, \eta_s)$ is the potential function at site s when it is allocated to class c. Throughout this paper the potential function is defined, $V(c, \eta) = \frac{1}{4} \sum_{t \in \eta} (c \oplus x_t)$, where $\oplus$ is an operator defined to take the value -1 if its arguments are equal, otherwise, $+1$.

Non-informative reference priors are chosen for the noise model parameters to ensure that the observed intensity data dominates any prior information incorporated into the model. Uniform priors are selected for $\{\mu_c, \beta^{(0)}_c; c \in \Lambda\}$ and k. For $\{\sigma_c, \forall c \in \Lambda\}$ the reference priors may be found using Jeffrey's formula for non-informative priors [10]. Normally these priors are improper, but here their range is restricted to facilitate normalisation and ensure that the model selection described later is valid.

3 MCMC Sampling from the Posterior Distribution

To obtain the MAP image segmentation via a stochastic relaxation algorithm an MCMC algorithm is used to sample from the posterior distribution.

The sampling scheme in this paper is based on the Gibbs Sampler [13] but Metropolis-Hastings sub-chains [11] are incorporated to enable the model parameters and number of classes to be sampled.

The sampling process comprises a sequential scan, updating the pixel sites and various model parameters in a specific order. The algorithm used here consists of the following moves:

(i) re-segment the image,
(ii) sample noise model parameters,
(iii) sample MRF model parameters,
(iv) sample the number of classes.

These moves are described in detail in the following sections of this paper.

3.1 Image Segmentation

The conditional density functions for the allocation of pixel s to class c may be derived from equation 5,

$$p(x_s = c \mid y_s, \eta_c, \Psi_c) \propto \frac{1}{\sqrt{2\pi\sigma_c^2 T_t}} \exp\left\{ -\frac{1}{T_t} \left[\frac{1}{2} \left(\frac{y_s - \mu_c}{\sigma_c} \right)^2 + \left(\beta_c^{(0)} + \beta^{(1)} \; V(c, \eta_s) \right) \right] \right\} \tag{6}$$

where T_t is the annealing temperature at iteration t of the algorithm. Gibbs sampling is used to update the allocation of class labels to pixel sites using the above conditional density function.

3.2 Sampling Noise and MRF Model Parameters

Metropolis-Hastings samplers are used to update noise and MRF model parameters. The proposal densities used are zero mean Gaussian with variances dependent on the parameter being sampled. The conditional density function for the noise model parameters for class c is,

$$
\begin{aligned}
p(\mu_c, \sigma_c \mid Y,\ X) &\propto \prod_{s:x_s=c} p(y_s \mid \mu_c, \sigma_c)\ p_r(\mu_c) p_r(\sigma_c) \\
&= \frac{1}{\sigma_c\ (2\pi\sigma_c^2 T_t)^{n_c}} \exp\left\{ -\frac{1}{2T_t} \sum_{s:x_s=c} \left(\frac{y_s - \mu_c}{\sigma_c} \right)^2 \right\}
\end{aligned}
\tag{7}
$$

where $n_c = \#(s : x_s = c)$. The conditional probability density functions for the external field model parameters, $\{\beta_c^{(0)}, c \in \Lambda\}$ are given by,

$$
\begin{aligned}
&p(\beta_c^{(0)}, c \in \Lambda \mid X) \\
&\quad = \ p(X \mid \beta_c^{(0)}, c \in \Lambda)\ p_r(\beta_c^{(0)}, c \in \Lambda) \\
&\quad = \prod_{(c\in\Lambda,\forall\eta)} \left(\frac{\exp\left(-\frac{1}{T_t}[\beta_c^{(0)} + \beta^{(1)}\ V(c,\eta)]\right)}{\sum_{i\in\Lambda} \exp\left(-\frac{1}{T_t}[\beta_i^{(0)} + \beta^{(1)}\ V(i,\eta)]\right)} \right)^{n_{(c,\eta)}} \\
&\quad = \prod_{(c\in\Lambda)} \left(\frac{\exp\left(-\frac{1}{T_t}[\beta_c^{(0)}]\right)}{\sum_{i\in\Lambda} \exp\left(-\frac{1}{T_t}[\beta_i^{(0)}]\right)} \right)^{n_c} \\
&\quad \times \prod_{(c\in\Lambda,\forall\eta)} \left(\frac{\exp\left(-\frac{1}{T_t}[\beta^{(1)}\ V(c,\eta)]\right) \sum_{i\in\Lambda} \exp\left(-\frac{1}{T_t}[\beta_i^{(0)}]\right)}{\sum_{i\in\Lambda} \exp\left(-\frac{1}{T_t}[\beta_i^{(0)} + \beta^{(1)}\ V(i,\eta)]\right)} \right)^{n_{(c,\eta)}}
\end{aligned}
\tag{8}
$$

where $n_{(c,\eta)} = \#(s : x_s = c, \eta_s = \eta)$. If the parameters $\beta_i^{(0)}$ are normalised so that, $\sum_{i\in\Lambda} exp(-\beta_i^{(0)}) \ = \ 1$, then the term $exp(-\beta_i^{(0)})$ becomes analogous to a weight term in a mixture density. By introducing this dependency between $\beta_i^{(0)}$ parameters it becomes possible to maintain a balance between the external field strength and the strength of interaction between neighbouring pixels. This makes it possible to set the pixel interaction strength *a priori*. The introduction of this weight term also proves highly convenient for proposing reversible jumps, as will be seen in the following section.

It is theoretically possible to sample the $\beta^{(1)}$ parameter from the conditional density,

$$
\begin{aligned}
p(\beta_c^{(1)}, c \in \Lambda \mid X) \ &= \ p(X \mid \beta_c^{(1)}, c \in \Lambda)\ p_r(\beta_c^{(1)}, c \in \Lambda) \\
&= \prod_{(c\in\Lambda,\forall\eta)} \left(\frac{\exp\left(-\frac{1}{T_t}\beta_c^{(1)}\ V(c,\eta)\right)}{\sum_{i\in\Lambda} \exp\left(-\frac{1}{T_t}[\beta_i^{(0)} + \beta^{(1)}\ V(i,\eta)\right)]} \right)^{n_{(c,\eta)}}
\end{aligned}
\tag{9}
$$

Unfortunately, under particular underlying label map configurations the posterior density for $\beta^{(1)}$ will not be proper.

$$
\lim_{\beta^{(1)}\to-\infty} p(X = x', \Psi \mid Y = y) \ = \ \alpha \tag{10}
$$

where α is a positive constant. This is a direct consequence of approximating the likelihood by the pseudo-likelihood. It may be possible to overcome this problem by choosing a suitable prior but for simplicity, the results presented in section 4 of this paper are obtained with $\beta^{(1)}$ given *a priori*.

3.3 Reversible Jumps to sample the Number of Classes

Reversible jumps were developed in [12] to allow a Metropolis-Hastings sampler to sample from a variable sized class space, $\Lambda = \{0, 1, ..., k\}$, i.e. where k changes. When k increases the model parameters associated with the new class are derived from the old parameter values and random variables. This allows the dimension of the parameter vector to be preserved.

If two models are considered, labelled $m1$ and $m2$, and $\Theta^{(1)}$ and $\Theta^{(2)}$ are their associated parameter vectors of dimension n_1 and n_2 respectively, then to jump between their parameter spaces requires the generation of random vectors $u^{(1)}$ and $u^{(2)}$ such that, $\#(\Theta^{(1)}) + \#(u^{(1)}) \;=\; \#(\Theta^{(2)}) + \#(u^{(2)})$, where $\#(\cdot)$ indicates the dimension of the inclosed vector.

If y represents observed data, the acceptance ratio based on sampling the posterior distribution, for the transition of $x = (m1, \Theta^{(1)})$ to $x' = (m2, \Theta^{(2)})$ is derived in [12],

$$min\left\{1, \frac{p(2, \Theta^{(2)})\; p(1, \Theta^{(1)} \mid 2, \Theta^{(2)})\; p(u^{(1)})}{p(1, \Theta^{(1)})\; p(2, \Theta^{(2)} \mid 1, \Theta^{(1)})\; p(u^{(2)})} \;\left|\frac{\partial(\Theta^{(2)}, u^{(2)})}{\partial(\Theta^{(1)}, u^{(1)})}\right|\right\} \qquad (11)$$

where $\left|\frac{\partial(\Theta^{(2)}, u^{(2)})}{\partial(\Theta^{(1)}, u^{(1)})}\right|$ is the Jacobian determinant for the mapping functions of the transformation of $[\Theta^{(1)}, u^{(1)}]$ to $[\Theta^{(2)}, u^{(2)}]$.

This methodology was applied in [14] to the problem of fitting a Gaussian mixture model with an unknown number of components to observed data. The posterior distribution was Gibbs sampled with sequential sweeps, each of which allocated classes, updated model parameters and a reversible jump was incorporated to update the number of mixture components, k. At each sweep the algorithm would randomly choose between proposing to increment or decrement the number of classes.

If the number of classes was to be incremented then a class was randomly chosen to be split and new parameters generated for each of the two new classes. The observed data samples allocated to the splitting class were reallocated using a Metropolis Sampler based on an acceptance ratio incorporating their conditional density functions.

If the number of classes was to decrease, two classes that were in some relevant way adjacent, were chosen to be merged. New parameters for the merged class were then generated from the two old sets of parameters. All data samples allocated to either of the two classes being split were then automatically assigned to the new merged class with probability one.

The problem of splitting a region of a Markov Random Field labelled by a single class, c, into a region composed of two classes, c_1, c_2, is similar to the splitting of a component of the mixture distributions described in [14]. Hence a similar approach may be followed. Each class is defined by a set of parameters, $\Psi_c = \{\mu_c, \sigma_c, \beta_c^{(0)}\}$. When splitting one class into two, three new parameters need to be created and the values of all six can then be set with three degrees of freedom. To preserve the dimensionality of the parameter space three random variables are generated, u_1, u_2, u_3. So when splitting state c into c_1 and c_2 a transform between $[\mu_c, \sigma_c, \beta_c^{(0)}, u_1, u_2, u_3]$ and $[\mu_{c1}, \sigma_{c1}, \beta_{c1}^{(0)}, \mu_{c2}, \sigma_{c2}, \beta_{c2}^{(0)}]$ has to be defined.

The three parameters for each of the two new classes may be derived from the old values, $\mu_c, \sigma_c, \beta_c^{(0)}$, and the random variables u_1, u_2, u_3. The new parameters are calculated to preserve the 0th, 1st and 2nd order moments across the transformation. The resulting values given by the following set of equations:

$$\begin{array}{lcl@{\qquad}lcl}
\beta_{c1}^{(0)} & = & \beta_c^{(0)} \; - \; T_t \; ln(u_1) & \beta_{c2}^{(0)} & = & \beta_c^{(0)} \; - \; T_t \; ln(1-u_1) \\
\mu_{c1} & = & \mu_c - u_2 \; \sigma_c \; \sqrt{\frac{1-u1}{u1}} & \mu_{c2} & = & \mu_c + u_2 \; \sigma_c \; \sqrt{\frac{u1}{1-u1}} \\
\sigma_{c1}^2 & = & u_3 \; (1-u_2^2) \; \sigma_c^2 \; \frac{1}{u_1} & \sigma_{c2}^2 & = & (1-u_3) \; (1-u_2^2) \; \sigma_c^2 \; \frac{1}{1-u_1}
\end{array} \qquad (12)$$

The choice of random variables for, u_1, u_2, u_3, must be such that, $\{u_1, u_2, u_3 \in (0, 1]\}$. For this reason and to allow a bias towards splitting the data into roughly equal partitions, beta distributions are used to sample u_1, u_2, u_3.

The Jacobian determinant of these mapping functions, needed in the calculation of the acceptance ratio is,

$$\left| \frac{\partial(\Psi_{c1}, \Psi_{c2})}{\partial(\Psi_c, u_1, u_2, u_3)} \right| \; = \; \frac{T_t \sigma^2}{u_1^2 (1-u_1^2) \; \sqrt{u3(1-u3)}} \qquad (13)$$

The pixel sites allocated to the class or classes selected in the split or merge reversible jump need to be reallocated on the basis of the new parameters generated. If a merge is being proposed, then all sites allocated to the two old classes are relabelled with the new merged class, with probability one.

The difficulty occurs when splitting one class into two. If a reasonable probability of acceptance is to be maintained, the proposed reallocation of the sites needs to be completed in such a way as to ensure the posterior probability of that particular segmentation is relatively high. To achieve this it would be desirable to propose a reallocation of pixels using the conditional density given in equation 6. Unfortunately, this is not possible since the allocation of classes in the neighbourhood η_s has not yet been calculated.

By using a distance measure based on the new class parameters and the observed grayscale values of the surrounding region of pixels it is possible to make a ‘guess’ to which class the neighbouring pixels might be allocated. The measure used here has a precedent in this type of algorithm: the Kolmogorov-Smirnov distance was used in [5] to allocate pixel sites between classes based on grayscale values or transformations of grayscale, indicative of texture type.

The Kolmogorov-Smirnov distance is a measure of the closeness of two distribution functions. It may be applied to two samples of data to ascertain whether they have been drawn independently from the same distribution. If $\hat{F}_1(k)$ and $\hat{F}_2(k)$ are two independent sample distribution functions (i.e. histograms) defined,

$$\hat{F}(k) \;=\; \frac{1}{n}\,\#(i : y_i \leq k) \tag{14}$$

where n is the number of data samples, y_i so that $1 \leq i \leq n$. Then the Kolmogorov-Smirnov distance is the maximum difference between distributions over all k,

$$d(y^{(1)}, y^{(2)}) \;=\; \max_k \left|\hat{F}_1(k) - \hat{F}_2(k)\right| \tag{15}$$

The Kolmogorov-Smirnov distance is a useful for two reasons; its value is independent of the underlying distribution function and it is unaffected by outlying data values.

To propose a likely neighbourhood configuration so that the conditional probability in equation 6 may be calculated, the Kolmogorov-Smirnov distance between the distributions of pixel grayscales in windows centered on each of the pixels comprising the neighbourhood and the distribution functions of the classes to which they might be allocated is calculated. The class giving the smallest distance at each of the neighboring pixel sites is then used as the best 'guess' for substitution in equation 6.

The two properties described above ensure the suitability of this measure because its value will be independent of the class distribution functions and its insensitivity to outlying pixel values means it will closely model the interactions of a Gibbs MRF.

The acceptance ratio for splitting region c into $c1$ and $c2$, thus increasing the number of classes from k to k^+, is given by,

$$\begin{aligned} &\frac{p(X = x^+, \psi^+, k^+ \mid Y = y)}{p(X = x, \psi, k \mid Y = y)} \; \frac{1}{p_\beta(u_1)p_\beta(u_2)p_\beta(u_3)} \\ &\quad\times\; \frac{1}{p(segmentation)} \left|\frac{\partial(\Psi_{c1}, \Psi_{c2})}{\partial(\Psi_c, u_1, u_2, u_3)}\right| \end{aligned} \tag{16}$$

where, $p(X = x, \psi, k \mid Y = y)$ is the approximation to the posterior density defined by equation 5, $p_\beta(u)$ is the probability of proposing the random variable u, $p(segmentation)$ is the probability of the segmenting region c into $c1$ and $c2$ $|\cdot|$ is the Jacobian determinant given by equation 13. The acceptance ratio for the jump combining two states into one is simply the inverse of that given in equation 16. In this case $p(segmentation)$ is found in retrospect and u_1, u_2, u_3 are calculated by back-substitution into equation 12.

It is therefore possible to incorporate into the algorithm any of the information criteria used is [4] [3] by adding their compensatory terms as priors on model order k to the posterior probabilities of the reversible jump acceptance ratio.

4 Experimental Results

The segmentation algorithm, described in the previous sections has been applied to various computer synthesised mosaics. A geometric annealing temperature schedule was chosen so that the algorithm would converge in a reasonable time. The schedule is given by,

$$T_t \;=\; (1+\alpha_1)^{\alpha_2(1-\frac{t}{N_t})} \tag{17}$$

where t is the iteration number, N_t is the total number of iterations, and α_1 and α_2 are constants (which were arbitrarily set to 1.1 and 10.0, respectively).

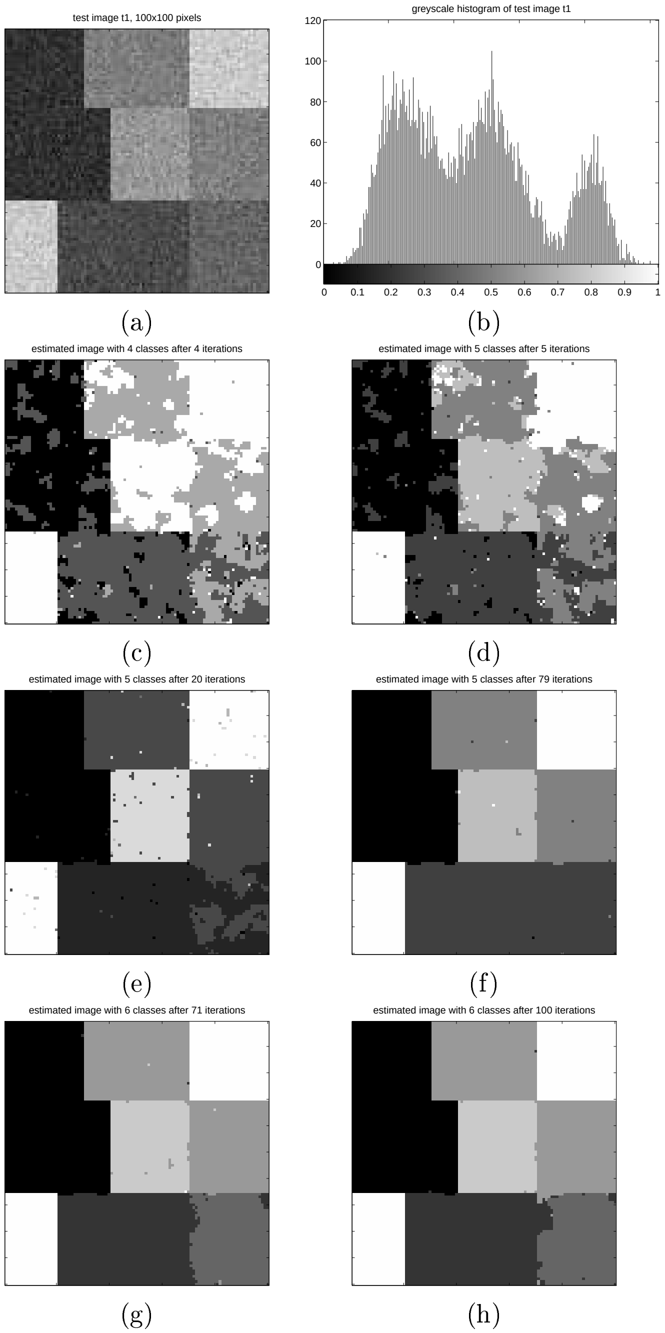
For the purposes of these experiments, $\beta^{(1)}$, which prescribes the overall strength of interactions between pixels within the image, was set *a priori* to a value of 1. 5. In proposing a reversible jump, the Kolmogorov-Smirnov distances for were calculated using 9×9 windows to generate 40 bin histograms of the pixel grayscale distribution functions.

Figure 1 shows the convergence of a 100 iteration run on a 6 class image. The segmentation, although not perfect is a good representation of the image after so few iterations and the number of classes has been correctly diagnosed. The results of a 500 iteration run are shown in figure 2. The original grayscale densities of the five classes are far closer than in the previous example (as shown by the grayscale histogram) but the algorithm has still converged to a good estimate of the underlying image.

A greyscale image of a house is segmented in Figure 3. The algorithm was repeated from various starting conditions and the results reached remained consistent. It could be argued that a better segmentation was arrived at in the intermediate steps but the algorithm then fits to a statistically if not visually better model in the final iterations.

5 Conclusion

In this paper we have presented an unsupervised segmentation algorithm for noisy images. The class model parameters, number of classes and pixel labels are all directly sampled from the posterior distribution using an MCMC algorithm. A single parameter is defined prior to the data (or at least an informative prior distribution needs to be defined) which defines the overall strength of neighbouring pixel interactions within the image. Excepting this, the algorithm is completely unsupervised. Experiments results have been presented in which synthesised images have been rapidly segmented to give accurate estimate of the original underlying image.



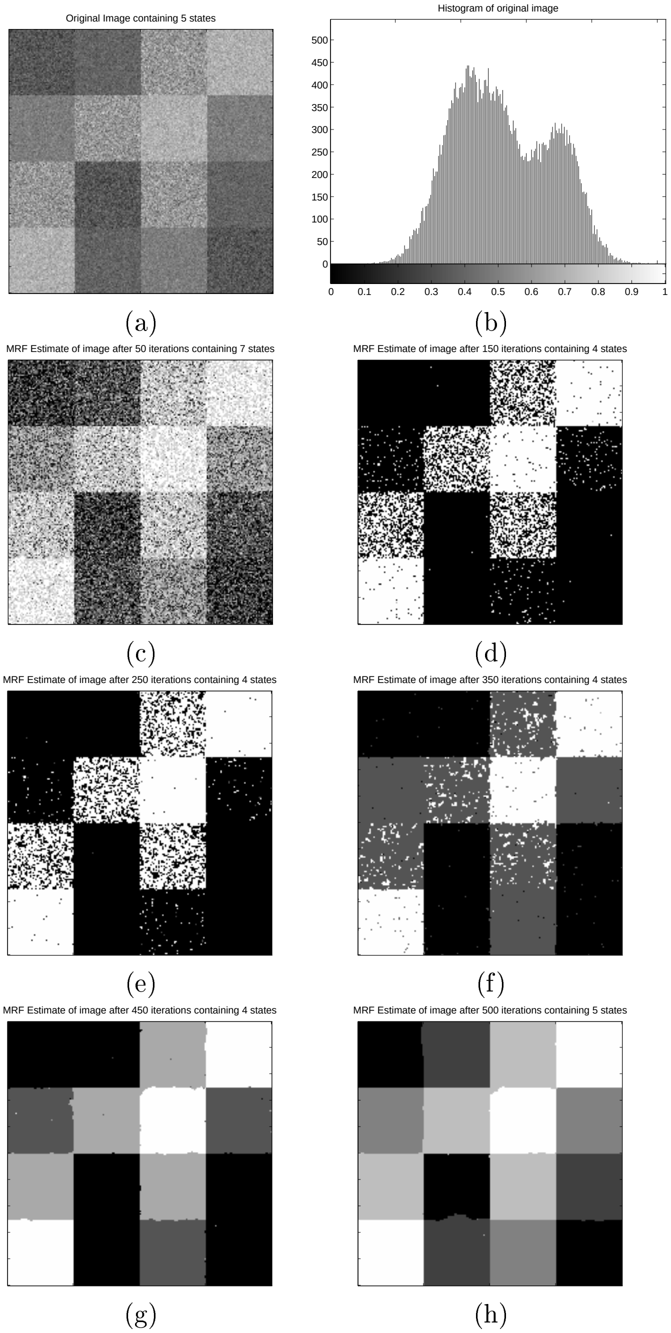
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 1: 100 iteration experiment on an image with six classes and beginning the algorithm with an arbitrary initial guess of four classes



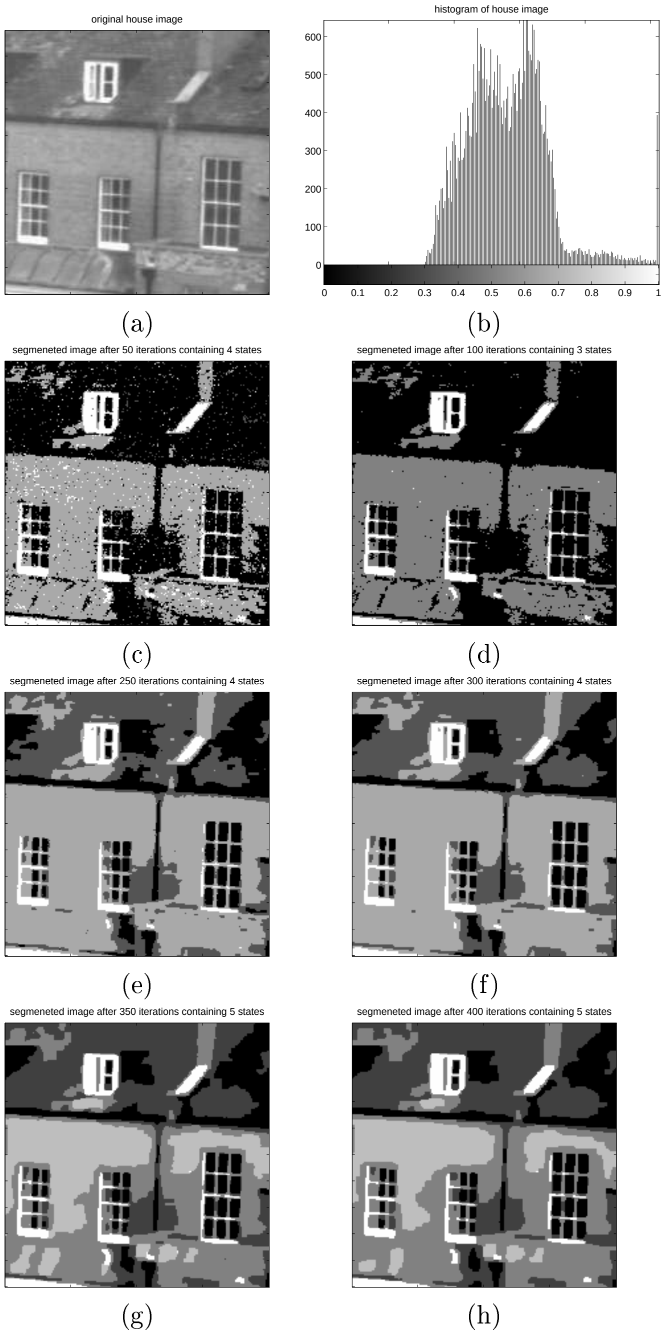
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 2: 500 iteration experiment on an image with five classes and beginning the algorithm with an arbitrary initial guess of six classes



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3: 400 iteration experiment on a greyscale image of a house. The algorithm begins with an arbitrary initial guess of six classes and consistently converges to five.

References

[1] B.S.Manjunath and R.Chellappa. Unsupervised texture segmentation using Markov Random Fields. *IEEE Trans. Patt. Anal & Machine Intell.*, 13(5):478–482, May 1991.

[2] C.Kervrann and F.Heitz. A Markov Random Field model-based approach to unsupervised texture segmentation using local and global statistics. *IEEE Trans. Image Processing*, 4(6):856–862, June 1995.

[3] C.S.Won and H.Derin. Unsupervised Segmentation of Noisy and Textured Images using Markov Random Fields. *CVGIP:Graphical Models and Image Processing*, 54(4):308–328, Jult 1992.

[4] J.W.Modestino D.A.Langan and J.Zhang. Cluster Validation for Unsupervised Stochastic model-based Image Segmentation. *Proc. ICIP94*, pages 197–201, 1994.

[5] C.Graffigne D.Geman, S.Geman and P.Dong. Boundary Detection by Constrained Optimization. *IEEE Trans. Patt. Anal & Machine Intell.*, 12(7):609–628, July 1990.

[6] D.K.Panjwani and G.Healey. Markov Random Field models for unsupervised segmentation of textured color images. *IEEE Trans. Patt. Anal & Machine Intell.*, 17(10):939–954, Oct 1995.

[7] F.S.Cohen and Z.Fan. Maximum Likelihood unsupervised texture segmentation. *CVGIP:Graphical Models and Image Processing*, 54(3):239–251, May 1992.

[8] H.H.Nguyen and P.Cohen. Gibbs Random Fields, Fuzzy Clustering, and the unsupervised segmentation of images. *CVGIP:Graphical Models and Image Processing*, 55(1):1–19, Jan 1993.

[9] J.Besag. On the statistical analysis of dirty pictures. *J. Royal Statist. Soc., Series B*, 48:259–302, 1986.

[10] J.M.Bernardo and A.F.M.Smith. *Bayesian Theory*. Wiley, 1994.

[11] L.Tierny. Markov Chains for exploring posterior distributions. *Annals of Statistics*, 22(5):1701–1762, 1994.

[12] P.J.Green. Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1996.

[13] S.Geman and D.Geman. Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Trans. Patt. Anal & Machine Intell.*, 6(6):721–741, Nov 1984.

[14] S.Richardson and P.J.Green. On Bayesian analysis of mixtures with an unknown number of components. -, Feb 1996.