# Parallel chains, delayed rejection and reversible jump MCMC for object recognition

M. A. Harkness & P. J. Green
School of Mathematics
University of Bristol
University Walk
Bristol, BS8 1TW, UK.
M.A.Harkness@Bristol.ac.uk

**Abstract**

We tackle the problem of object recognition using a Bayesian approach. A marked point process [1] is used as a prior model for the (unknown number of) objects. A sample is generated via Markov chain Monte Carlo (MCMC) techniques using a novel combination of Metropolis-coupled MCMC (MCMCMC) [2] and the Delayed Rejection Algorithm (DRA) [4]. The method is illustrated on some synthetic data containing simple geometrical objects.

## 1 Introduction

We address the problem of object recognition on a scene containing an unknown number of objects. Our approach is Bayesian using a marked point process [1] as a prior model for the (unknown number of) objects. In a marked point process each object is described by a pair $(l, m)$, where $l$ denotes a location and $m$ a vector of marks. The vector of marks contains information governing attributes such as type, size or texture. Our objective is to make inference about the number of objects and their characteristics. A similar approach was used by [9] in which ordered sets of vertices were used to describe an object's shape. We choose a simpler shape model using simple geometrical objects, rectangles and ellipses. The appeal of these models lies in the small number of parameters required to describe the shape. We face difficulties in sampling from a particularly peaked multimodal posterior distribution. We propose a method for overcoming such problems.

### 1.1 Image model

Our data are noisy images. Each is discretised into a lattice of square pixels, of $R$ rows and $C$ columns. The centres of each pixel are represented by the coordinates $(r, c)$ for observations $y_{rc}$. Let $U$ represent the set of possible objects. We denote each object by $x_i$ where $i = 1, ..., k$ and $k$ is the number of objects. Any configuration of objects can then be described by an unordered set $\mathbf{x} = \{x_1, x_2, ..., x_k\}$. Let $\Lambda$ denote the image space $[\Leftrightarrow 0.5, C \Leftrightarrow 0.5] \times [\Leftrightarrow 0.5, R \Leftrightarrow 0.5]$. Despite the image being discretised horizontal and vertical locations of an object are measured on a continuous scale over the ranges $[\Leftrightarrow 0.5, C \Leftrightarrow 0.5]$ and $[\Leftrightarrow 0.5, R \Leftrightarrow 0.5]$ respectively. The location of an object is its centre which we denote by $\xi_i = (\xi_i^h, \xi_i^v)$, for object $i$, with the superscripts $h$ and $v$ denoting the

BMVC 2000 doi:10.5244/C.14.73

horizontal and vertical locations respectively. We assume that our data are produced by the following model,

$$y_{rc} = \mu_{rc} + \epsilon_{rc} \quad r = 0, ..., R \Leftrightarrow 1; \qquad c = 0, ..., C \Leftrightarrow 1;$$

where $\epsilon_{rc}$ i.i.d. $N(0, \sigma^2)$. We regard $\mu_{rc}$ as the true value of the intensity in pixel $(r, c)$.

We assume that the true scene has a foreground and background whose intensities we denote by $\mu^f$ and $\mu^b$ respectively. A pixel is regarded as lying in the foreground if it lies inside an object, i.e. its centre lies inside an object. Otherwise the pixel is regarded as lying in the background. Then

$$\mu_{rc} = \begin{cases} \mu^f & \text{if pixel } (r,c) \text{ lies in the foreground} \\ \mu^b & \text{if pixel } (r,c) \text{ lies in the background} \end{cases} . \tag{1}$$

This model has been used in several image processing applications, notably by [7], [9] and [8].

## 1.2 Model for an unknown number of objects

Using a marked point process [1] each object $x_i$ is represented by a point giving its location and attached to each object is a vector of marks. The marks are assumed to lie in some space $\Omega$, thus the object $x_i$ lies in $U = \Lambda \times \Omega$ and any configuration of objects $\mathbf{x}$ is a marked point process on $U$. We model the distribution of $\mathbf{x}$ by means of its density $\pi(\mathbf{x})$ relative to a basic reference process, which is a Poisson object process on $U$ with intensity $u = \lambda \times \omega$ where $\lambda$ is the Lebesgue measure on $\Lambda$ and $\omega$ is a probability measure on $\Omega$. Then under the Poisson object process on $U$ the total number of objects follows a Poisson distribution with mean $u(U)$. Further, given that there are $k$ objects, these objects are i.i.d. distributed in $U$ with distribution $u/u(U)$.

To model interactions we use a pairwise interaction model for the joint density of $\mathbf{x} = \{x_1, ..., x_k\}$,

$$\pi(\mathbf{x}) \propto \gamma^k \prod_{i<j} f(x_i, x_j). \tag{2}$$

The density (2) is a model for an unknown number of objects. In this model $\gamma > 0$ is a constant, $k$ is the number of objects and $f : U \times U \to [0, \infty)$ is the interaction function. For any fixed choice of $f$, provided $f$ is well-defined and integrable, the value of $\gamma$ can be chosen to encourage or penalise the number of objects $k$ in the configuration $\mathbf{x}$.

Now let $\sim$ be any symmetric, reflexive binary relation on $U$. We specify an interaction function $f$ by reference to the relation $\sim$. The binary relation $\sim$ can be defined as

$$x_i \sim x_j \Leftrightarrow R(x_i) \cap R(x_j) \neq \emptyset, \tag{3}$$

where $R(x_i)$ represents some region of the image connected with object $i$. $R$ and the interaction function $f$ can be chosen to exhibit the kind of spatial interaction present. We take $R(x_i)$ to represent the spatial extent of object $i$ and define $f$ as

$$f(x_i, x_j) = \begin{cases} 0 & \text{if } x_i \sim x_j \\ 1 & \text{otherwise} \end{cases} , \tag{4}$$

thus forbidding objects to overlap. This model is known as the hard-core object process.

### 1.3 Mark distributions

As we are concerned with rectangles and ellipses only two parameters are required to describe an object's shape. Thus the set of marks contains parameters for type, horizontal and vertical size, orientation and intensity. The object type is denoted by a categorical variable $\tau_i$ for object $i$. We assign equal probability to each object type in the type mark density.

The horizontal and vertical size parameters are denoted by $\psi_i^h$ and $\psi_i^v$ respectively. The actual horizontal and vertical sizes of an object in pixels are given by $\psi_i^h \delta^h$ and $\psi_i^v \delta^v$. The constants $\delta^h$ and $\delta^v$ are the standard horizontal and vertical sizes and must be specified in connection with the particular image data. In the applications we consider we assume $\psi_i^h$ and $\psi_i^v$ have maximum and minimum values denoted by $\underline{\psi^h}$, $\overline{\psi^h}$ and $\underline{\psi^v}$, $\overline{\psi^v}$ respectively. These values are assumed known. We choose a truncated gamma density as prior for $\psi_i^h$ (for both object types) with parameters $\alpha^h$ and $\beta^h$ parameterized so that $\alpha^h/\beta^h$ and $\alpha^h/{\beta^h}^2$ are the mean and variance, respectively, before truncation. There is an additional hierarchical level of our model to account for variation in object size. We let the parameter $\beta^h$ follow a gamma distribution with parameters $\epsilon^h$ and $\varepsilon^h$ (again adopting the parameterisation in which $\epsilon^h/\varepsilon^h$ is the mean and $\epsilon^h/{\varepsilon^h}^2$ is the variance). The set up for vertical components is identical.

The prior on $\theta_i$, the orientation, is taken to be $U(0,\pi)$ for both object types since the object shapes we consider are identical under rotations through $\pi$. For the intensity parameter $\mu_i$ we choose a normal distribution with mean $\lambda$ and variance $\nu^2$. The Bayesian hierarchical model is displayed in Figure 1 as a Directed Acyclic Graph (DAG).

## 2 Simulation from the posterior

The Bayesian approach makes use of the posterior density which is given by

$$\pi(\mathbf{x}|\mathbf{y}) \propto \pi(\mathbf{x})\pi(\mathbf{y}|\mathbf{x}), \tag{5}$$

to make inference about the underlying scene, $\mathbf{x}$, given the data, $\mathbf{y}$. The posterior comprises prior and likelihood models which are respectively used to reflect our beliefs about the true scene and to model the noise. Our prior model is made up of (2) and the mark distributions. The likelihood is assumed to be additive Gaussian noise

$$\pi(\mathbf{y}|\mathbf{x}) \propto \prod_{\substack{r=0,\ldots,R-1 \\ c=0,\ldots,C-1}} \exp\left\{ \frac{\Leftrightarrow(y_{rc} \Leftrightarrow \mu_{rc})^2}{2\sigma^2} \right\}. \tag{6}$$

We make the assumption that the noise variance, $\sigma^2$, is known. We have a good estimate for $\sigma^2$ since we consider some artificial data for which we know the noise level. However, in many real situations the noise is unknown and needs to be estimated. An alternative which is within the scope of our modeling approach is to include an additional hierarchical layer for $\sigma$ and estimate $\sigma$ from the simulation.

In many Bayesian problems the posterior is only known up to proportionality (as is the case here) and direct evaluation of (5) is impossible. We make inference based on a sample obtained via Markov chain Monte Carlo (MCMC) techniques in which our target distribution is (5).

## 2.1 MCMC sampling techniques

MCMC techniques originally appeared in the statistical physics literature and have become popular in recent years in statistics, particularly with the increase in computational power. The Metropolis algorithm [6] proceeds by proposing some new value for a variable of interest. A decision on whether to accept the proposed new value is based upon a ratio $R$. The proposal is accepted with probability

$$\min\{1, R\}, \tag{7}$$

otherwise we retain the current value. Extensions of this algorithm proceed in the same proposal-accept/reject fashion using an expression of the same form as (7). Of particular interest is the reversible jump algorithm of [3] for variable dimension problems and the Delayed Rejection Algorithm (DRA) of [4] which allows additional moves to be attempted should our original move be rejected.

In addition we consider Metropolis-coupled MCMC (MCMCMC) which was proposed by [2]. The idea is to run $n$ parallel chains each with a different target distribution. Let us denote the target distributions by $\pi_i(x)$, $i = 1, ..., n$. We set $\pi_1(x) = \pi(x)$ and choose $\pi_2(x), ..., \pi_n(x)$ to improve mixing. A complete sweep is made of all move types in each chain. We then attempt to swap the states of two chains using the Metropolis-Hastings algorithm [5]. Inference is based solely on output from chain $\pi_1(x)$.

When implementing MCMCMC we need to specify a family of distributions $\pi_i(x)$, $i = 1, ..., n$. Our preference is some family of the form

$$\pi_i(x|y) \propto \pi(x)\pi_i(y|x), \tag{8}$$

where the effect of the likelihood is altered in different chains by some temperature scheme. Using (8) also allows us to retain prior information regarding the size and number of objects in any configuration. Such a choice ensures that the prior terms disappear from the ratio $R$ in (7) making the move relatively simple to implement.

## 2.2 Sampling from the posterior

The successful and reliable application of MCMC requires that we achieve satisfactory mixing over all of the parameters in the parameter space. In our application this means that we move freely between different configurations of objects and do not become stuck in some local mode nor to move too slowly through the parameter space. The sampling procedure may be regarded as the fitting of all possible configurations of objects (as defined by our model) to the data and the accumulation of votes for each of these with those visited more often deemed to be more probable representations of the data. It is therefore imperative that good mixing is attained if reliable inference is to be made.

In the application we consider the posterior distribution is exceedingly peaked and multimodal. Sampling from the posterior proves to be extremely difficult because of this and should a naive sampling approach be adopted we find the sampler moves about very slowly and has difficulty moving between modes.

One sweep of our algorithm consists of attempts to update all of the model parameters. Note that the procedure could be carried out in parallel using $n$ processors to sample from each of the $n$ chains. The sampling procedure contains the following moves

(i) update locations,

(ii) update types,
(iii) update size scales,
(iv) update horizontal parameters,
(v) update vertical parameters,
(vi) update orientations,
(vii) update intensities,
(viii) split or merge,
(ix) birth or death,
(x) swap two configurations.

Moves (i) - (vii) use standard Metropolis-Hastings updates [5] and are carried out for each object. Proposals to alter the number of objects in moves (viii) and (ix) require the use of reversible jump MCMC [3]. Note that in both of these moves a random choice is made between proposing to increase or decrease the number of objects. In addition there is a DRA step applied to move (viii) in which we make a second attempt to split or merge should the first attempt fail.

After cycling through each chain and updating each of the parameters for each object and the number of objects we attempt to swap the configurations of two chains. This move uses MCMCMC [2] and the DRA [4]. Suppose $i$ and $j$ are the chains chosen for swapping. We define $a^{th}$ order adjacency between two chains if $|i \Leftrightarrow j| = a$. A chain $i$ is chosen at random and we propose in the first instance to swap its configuration with that from a chain $j$ which is chosen at random and satisfying at least $2^{nd}$ order adjacency, i.e. $|i \Leftrightarrow j| > 1$. Constructing the move in such a fashion allows us to make larger jumps and hence explore the posterior much quicker thus requiring fewer sweeps. Should the move fail then a second stage move is proposed. We retain the originally selected chain and now attempt to swap its configuration with that of a chain $k$ which is chosen at random and satisfying $1^{st}$ order adjacency, i.e. $|i \Leftrightarrow k| = 1$. Note that it is perfectly feasible to consider more stages in a bid to make larger jumps around the parameter space, although in our application we do not find this to be necessary.

# 3 Application to data

Our sampling procedure and model are applied to the noisy synthetic image in Figure 2. The true image contains 8 objects of which 5 are rectangles and 3 are ellipses. The image is square with 50 by 50 pixels. Note that the orientation and intensity parameters are fixed in this example. The sampler is run for 160000 iterations of which 80000 burn-in. We simulate 7 parallel chains.

Some output is displayed in Figures 3 and 4. First we comment on Figure 3. The histogram of the number of objects (a) shows that the sampler favours fewer than the true number of objects since the posterior mode is 5, the mean is 5.533 with standard deviation 1.276. We experience difficulty in finding the smaller objects and find that there are few occasions when the sampler finds multiple objects in the place of a true object. The boxplot (b) shows the spread of the number of objects.

Histograms for the number of rectangles and ellipses can be found in (c) and (d). Rather than considering these individually we look at the number of rectangles and ellipses when there are $k = 6$ objects, reflecting the posterior mean for $k$. The plot (e) is a

jittered scatter plot of the number of rectangles against the number of ellipses. The plot takes its name since the observations are jittered around the actual intersection point, each observation being placed at random in a square centred on the actual point of intersection. Clearly there are a large number of observations for some of the pairings and this makes it difficult to get a handle on the individual frequencies. We focus on the most probable pairings, which are easily determined through (f), for which we can see the distribution of rectangles and ellipses is based around 5 or 6 objects with strong posterior probability of 2 or 3 ellipses with 4 or 3 rectangles, see (g). The uncertainty over the types of the 6 objects is due to there being less of a difference between rectangles and ellipses the smaller the objects.

Figure 4 (a) contains samples from the posterior taken every 1000 iterations after burn-in. Despite the relatively small number of samples it is quite easy to observe the samplers behaviour. There is a lot of activity around the object in the top right hand corner and the three objects which lie in the bottom right hand part of the image indicating some uncertainty over the number of objects and their marks in theses areas. It can also be seen that successfully merges objects together on occasion. The two small objects we thought to have missed in many configurations can now be seen to be the small ellipse at the bottom right of the large rectangle and the small rectangle in the bottom left hand corner of the image. The samples indicate there is some variation in location, size and type suggesting that the sampler has been successful in exploring the parameter space. The remaining plots in Figure 4 contain samples drawn from the posterior for different values of $k$ and pixelwise posterior probabilities. The latter two plots (g) and (h) contain the pixelwise probability of a pixel being in the foreground and the pixelwise probability of misclassification (i.e. placing a pixel in the foreground when it in truth it lies in the background and vice versa). The areas of activity can be clearly seen from these two plots. In particular the misclassification of pixels around the edge of ellipses can be clearly seen.

As far as splitting and merging is concerned we find that the sampler does not visit configurations containing clusters very often unless there is some degree of uncertainty over what is actually there and the objects are not too large, as is the case with the objects in the bottom right corner of the image. Nevertheless this is to be expected since the shape model we have specified does not assign very high probability to two objects covering the same pixels as an existing object, nor is it particularly flexible. This raises an important point regarding shape models and implies that in the scenario we consider it may be preferable to consider using a more flexible shape model, although at the expense of an increased number of model parameters.

In complex problems such as this, where the dimensionality is high, it is essential to design efficient samplers if the parameter space is to be explored swiftly. Overall the sampling mechanism we have implemented has successfully allowed us to move between models and attain good mixing over $k$, which in turn allows reliable inference to be made. Varying the model parameters relating to the expected number of objects and the size of objects in the image can help us find the smaller objects and multiple objects where the true objects lie but only when we depart considerably from our prior beliefs about the true scene. For example, increasing $\gamma$ and forcing more objects into configurations, as well as decreasing $\delta$ proves more successful in finding all of the objects. One refinement of the sampler which may be considered is to change the birth and death move to propose the birth of smaller objects. We would expect this to lead to more new objects being introduced into configurations, and whilst this would not change the results of the simulation,

it may mean we explore the parameter space swifter and therefore require fewer iterations in total. The method is also quite robust in the presence of noise and can achieve satisfactory results simply by careful parameter choice of the $n$ chains. This is of particular interest since in many applications the noise level is unknown yet obtaining an accurate estimate of the noise is not crucial to the implementation of our sampler.

The method has been successfully applied in some other examples in which object size varies. Model variations, allowing the orientation and intensity parameters to vary, have also been successfully tackled. Recent work has focussed on a real problem on an image containing galaxies and stars in which the aim is automated classification. Consideration of this real problem in which the object shapes are more complex than in our synthetic example illustrates the success of the method. We expect the method to be useful in a variety of problems, not just image analysis. Further goals include the formulation of some sound implementation advice for general application of the sampler.

## References

[1] A. J. Baddeley and M. N. M. Van Lieshout. Stochastic geometry models in high-level vision. In K. V. Mardia and G. K. Kanji, editors, *Statistics and Images: Volume 1*, Journal of Applied Statistics, pages 233–258. Carfax, Abingdon, 1993.

[2] C. J. Geyer. Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163, 1991.

[3] P. J. Green. Reversible jump Markov chain Monte Carlo and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[4] P. J. Green and A. Mira. Delaying rejection in reversible jump Metropolis-Hastings. Technical Report S-99-01, Department of Statistics, University of Bristol, 1999.

[5] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[6] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[7] A. Pievatolo and P. J. Green. Boundary detection through dynamic polygons. *Journal of the Royal Statistical Society Series B*, 60:609–626, 1998.

[8] W. Qian, D. M. Titterington, and J. N. Chapman. An image analysis problem in electron microscopy. *Journal of the American Statistical Association*, 91:944–952, 1996.

[9] H. Rue and M. A. Hurn. Bayesian object identification. *Biometrika*, 86:649–660, 1999.

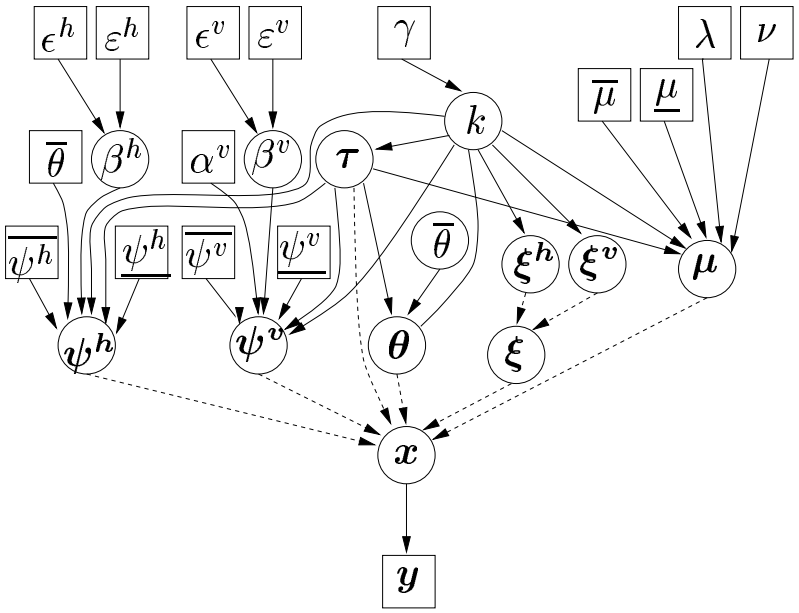


Figure 1: Directed Acyclic Graph (DAG) for the model formulated. In the DAG squares represent fixed quantities and circles unknowns. Solid lines denote deterministic relationships and dotted lines denote logical relationships.

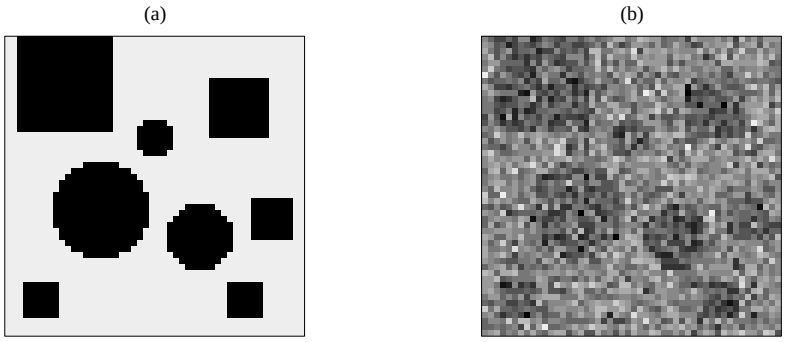


Figure 2: Synthetic image (a) and (b) noisy version.

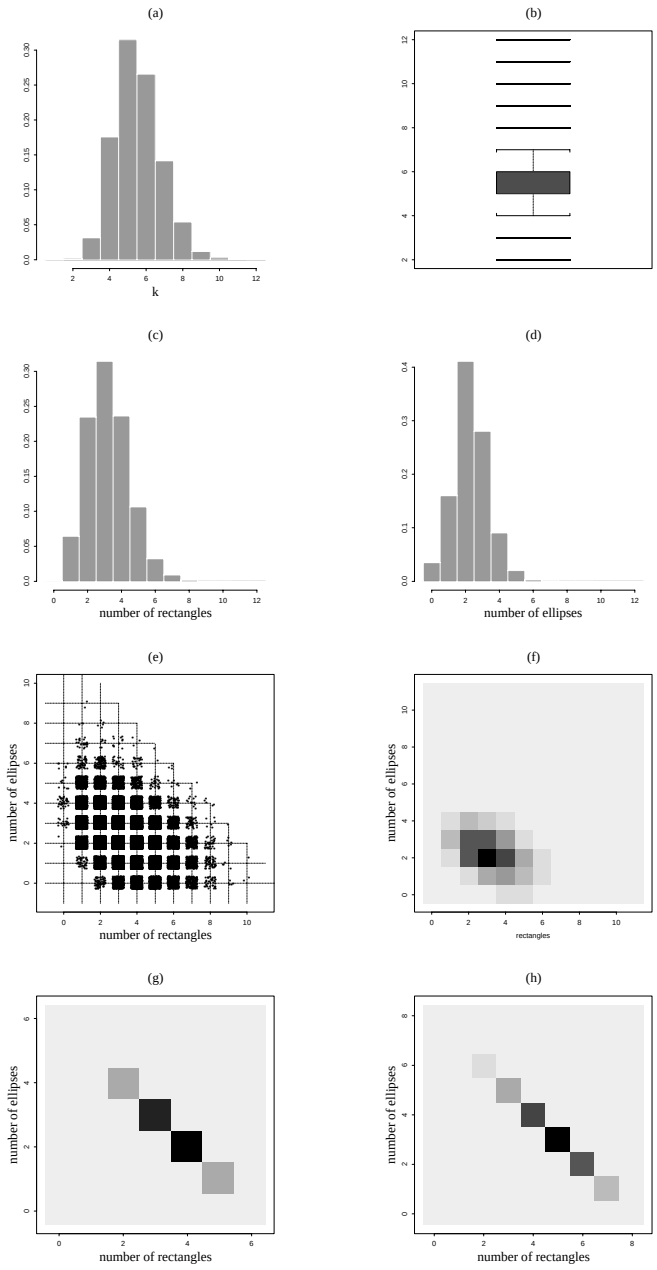


Figure 3: Posterior summary. (a) Histogram for $k$. (b) Boxplot for $k$. (c) Histogram for the number of rectangles. (d) Histogram for the number of ellipses. (e) Jittered scatterplot of ellipses and rectangles. (f) 2-d histogram of ellipses and rectangles. (g) 2-d histogram of ellipses and rectangles for $k = 6$. (h) 2-d histogram of ellipses and rectangles for $k = 8$.

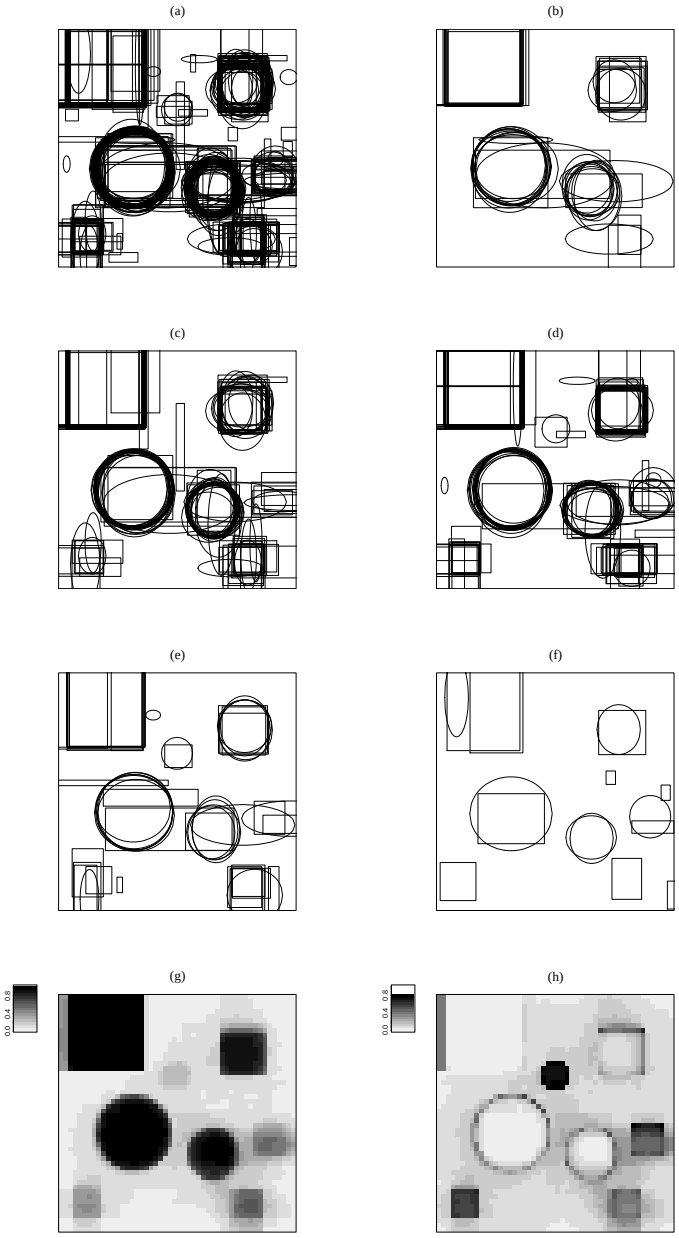


Figure 4: Analysis of synthetic image *A* with fixed orientation and intensity — posterior summary. Samples from the posterior (a) taken every 1000 iterations after burn-in. Of these samples we have (b) for $k = 4$, (c) for $k = 5$, (d) for $k = 6$, (e) for $k = 7$ and (f) for $k = 8$. Pixelwise posterior probability of (g) inclusion and (h) misclassification.