A. L. Reno[a], D. F. Gillies[b] and D. M. Booth[a]

[a]DERA, St Andrews Road, Malvern, Worcs, WR14 3PS, UK.

[b]Imperial College, 180 Queen's Gate, London SW7 2BZ, UK.

A new type of deformable model is presented that is able to combine some of the characteristics of both snakes and templates. It can be used to segment and recognise two–dimensional objects when only vague prior knowledge about their shapes is available. A jump-diffusion process is used to fit the template to the image. The jumps allows the template to undergo abrupt discontinuous changes in shape and position and to decide among multiple target models. The diffusion process allows the template to perform continuous flowing deformations like a snake. A prior shape model is described that uses the local and global characteristics of each different target class. An efficient form for the image likelihood is given that extends to multiple attributes and multiple images. The jump transition kernel defines the probabilities of the template jumping to a new state. This is difficult to generate and sample in practice though. To allow for this a method is described where a marginal transition kernel is generated by integrating over the continuous internal parameters for subsets of jumps. This makes the sampling problem much easier while still providing effective inferencing. The relation of this approach to active contours and region competition is discussed. It is shown that with the appropriate choice of prior and likelihood that snakes can easily be modelled within the deterministic part of the diffusion process. The method is demonstrated with the detection of buildings and planes in infrared and optical images and a comparison with an active contour is also given.

Keywords: Shape, deformable templates, jump-diffusion, snakes, Bayesian, fusion, aerial images

The problem considered here is the detection and classification of approximately located objects in two–dimensional aerial images. The intention is that this information will be used as a cue for more specialised three–dimensional vision systems.[1] In many cases image attributes alone are insufficient for object detection. Methods that do not use prior object information often give unreliable results. Prior information can be used in a very general way with active contours[2] or in a more specific way with deformable[3] or fixed models.[4] If a model is too specific however, then detection is limited to particular instances of objects which are known and where a good match can be found. Alternatively if a model is too general then detection may fail in cases where image attributes are ambiguous. In the type of imagery considered here there are common classes of objects such as buildings and planes. Although the objects in each class share many common characteristics, the level of variability makes it very difficult to define a model to cope with a sufficiently wide range of instances of these objects. In this work a new type of deformable model is presented that is able to combine some of the characteristics of both snakes and templates. A jump-diffusion process is used to fit the template to the image. The jumps allow the template to undergo abrupt discontinuous changes in shape and position and to decide among multiple target models. The diffusion process allows the template to perform continuous flowing deformations like a snake. In fact with the appropriate choice of model the deterministic part of the diffusion process is equivalent to snakes. The template jumps provide an essential higher level of inference above the diffusions however. They allow the template to sample configurations that are not reachable through diffusion which can only act by local refinement on continuous parameters. In the first section below the model for the template is defined. The next section describes the inferencing process using jump-diffusion. In the last section the approach is demonstrated with detection and recognition in some infrared and optical images and a comparison with an active contour model is given.

Other author information: (Send correspondence to A.L.R.)
A.L.R.: E-mail: alreno@dera.gov.uk
D.F.G.: E-mail: dfg@doc.ic.ac.uk
D.M.B.: E-mail: dmbooth@dera.gov.uk

There are several different types or classes of target in the scenes considered. Each of these classes of target have their own particular shape characteristics. A separate prior model for each class captures these shape characteristics. The template is allowed to choose dynamically which is the best prior model as it is fitted to the image. In turn the prior means that the template is predisposed or biased to take on certain shapes. The prior model associates a potential or relative probability with any given shape of the template. In Bayesian terms a prior model indicates how well the template fits our prior knowledge or assumptions. A given template that has a high prior probability in a particular class would therefore be typical of the types of shapes in that class. The image is the observed data, and is taken into account with a likelihood function which gives the probability of the observed image given that the template is in a particular configuration. A high probability in the likelihood therefore indicates that the template is a good match to the image. Suppose v represents the template as an ordered set of co-ordinates of a polygon where k is the number of co-ordinates and $\mathbf{l}$ associates the template with a particular class of target. Let $\boldsymbol{\mathcal{I}}$ be the image data with co-ordinates x and y. The prior $p\left(v\right)$ and likelihood $L\left(\boldsymbol{\mathcal{I}} \mid v\right)$ are then associated though Bayes' theorem. Since the probability of $\boldsymbol{\mathcal{I}}$ is constant for a given image it is only necessary to find the template v that maximises the prior and the likelihood. The densities are only taken as relative probabilities because the normalising constant for each is too difficult to compute in general. This is not usually a problem however. The prior and likelihood are defined below while the following section considers the problem of inferring the most likely template and its class.

The prior for each class measures how well the template agrees with prior shape expectations of that class. The priors used here are based on training data that consists of silhouettes of example objects. They use a local and a global shape measure. The local shape measure considers independently, the probability of each section of length ℓ vertices of the template v occurring in a shape from the class. The global shape measure is based on a feature vector of numerical shape descriptors for which a class mean and covariance matrix is computed from the training data.[5] The prior for target class $\mathbf{l}$ is the joint probability of these

$$\log p\left(v\right) = \lambda_1 \sum_{i=1}^{k} \log p\left(v_i^{\ell} \mid \mathbf{l}\right) + \lambda_2 \log p\left(v \mid \mathbf{l}\right) + I\left(v\right), \tag{1}$$

where λ_i are constants. The fist term is the local shape measure. This probability density is taken from a co-occurrence matrix of the combined frequencies of boundary angle sequences in the training class shapes. The co-occurrence matrix for each target class is computed up to some fixed sequence length. The probability of any given sequence up to this length can then be computed through the marginal probability density. The second term gives the probability that the global features from the template are from a shape from the class. This is taken as a multivariate normal distribution where $\bar{\mathbf{y}}_{\mathbf{l}}$ is a mean and $\mathbf{Q}$ is a covariance matrix for the class. The features are scaled to unit variance and translated to zero mean. The third term $I\left(v\right)$ is an indicator function which is zero when v is non self intersecting (simple) and is a nominal large negative value otherwise. This prior uses general shape information taken from training data. However a much more general type of prior such as minimum boundary length or tension and stiffness could have been used instead to give the template snake like behaviour for example. Alternatively a more specialised type of prior could have been chosen for rectangular building like structures. An example could be $-\sum_i \sin^2\left(2\theta_i\right)$, where θ_i is the angle at each vertex.

The likelihood gives a measure of the match of the template to the image. The image that is observed however is always a degraded version of an ideal underlying image. The likelihood generally embodies some form of noise model to allow for this. The measure used here is based on a joint probability of image region and edge attributes,

$$\log L\left(\boldsymbol{\mathcal{I}} \mid v\right) = \mu_1 \log L_r\left(\boldsymbol{\mathcal{I}} \mid v\right) + \mu_2 \log L_e\left(\boldsymbol{\mathcal{I}} \mid v\right), \tag{2}$$

where μ_i are constants, $\log L_r\left(\boldsymbol{\mathcal{I}} \mid v\right)$ is the region term and $\log L_e\left(\boldsymbol{\mathcal{I}} \mid v\right)$ is the edge term. The region likelihood is the joint probability that each pixel within the region enclosed by the template belongs to the target distribution and every other pixel belongs to the background distribution*. Let the pixel population in the target region follow a

*The background distribution refers to the whole population of pixels that do not belong to the given target type.

known distribution $\alpha_{\boldsymbol{\alpha}}$ and the pixel population in the background region follow a different known distribution $\alpha_{\overline{\boldsymbol{\alpha}}}$. Assuming the pixels are conditionally independent given the distribution then the joint log probability is

$$\log L_r\left(\boldsymbol{\mathcal{I}} \mid v\right) = \int\int_{\mathcal{R}} \log p\left(\boldsymbol{\mathcal{I}} \mid \alpha_{\boldsymbol{\alpha}}\right) dx\, dy + \int\int_{\overline{\mathcal{R}}} \log p\left(\boldsymbol{\mathcal{I}} \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right) dx\, dy, \tag{3}$$

where $\mathcal{R}$ is the region enclosed by the template and $\overline{\mathcal{R}}$ is a fixed region outside the template. This can be taken as the whole image excluding the region enclosed by the template. Equation 3 can be expressed identically as a boundary integral using Green's theorem (see Appendix A). The only non constant term in the resulting expression is the difference between the log probabilities over the target and background region and this depends only on the boundary of the template itself. This reduces it to the contour integral

$$\log L_r\left(\boldsymbol{\mathcal{I}} \mid v\right) = \oint_{\boldsymbol{\Gamma}} \left\{ \int_0^y \log \frac{p\left(\boldsymbol{\mathcal{I}} \mid \alpha_{\boldsymbol{\alpha}}\right)}{p\left(\boldsymbol{\mathcal{I}} \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right)} dy \right\} dx, \tag{4}$$

where $\boldsymbol{\Gamma}$ is the boundary of the template[†]. The log term is the Bayes factor[7] which measures the significance of each pixel under the hypothesis that it belongs to the target against the hypothesis that it belongs to the background. This term contributes little when there is no significant difference between the probabilities under the two hypotheses. The integral function in the brackets depends only on the image and the probability densities of the target and background pixel populations. These integrals are constant functions and can be computed before fitting the template. This allows the calculation of the full region likelihood at very little computational expense. In general each target type will have its own probability density and so both the prior and likelihood will depend on the target type. It is not necessary for the target and background densities to follow parametric distributions such as Gaussians. It is feasible for example that these densities could be taken from the output of another detection system.

An important part of our overall approach is the fusion of multiple image cues.[1] This is quite easy here because the image data $\boldsymbol{\mathcal{I}}$ could just as easily be a vector of attributes such that $\boldsymbol{\mathcal{I}} = (\boldsymbol{\mathcal{I}}_1, \boldsymbol{\mathcal{I}}_2, \ldots, \boldsymbol{\mathcal{I}}_n)$. These attributes could include range data, texture, or multiple images for example. The joint probability is then considered over all image attributes where each extra independent attribute provides an additional constraint. In practice it is reasonable to assume conditional independence between the attributes given the target type so that,

$$p\left(\boldsymbol{\mathcal{I}}_i \mid \mathbf{l}, \boldsymbol{\mathcal{I}}_{-i}\right) = p\left(\boldsymbol{\mathcal{I}}_i \mid \mathbf{l}\right), \tag{5}$$

where $\boldsymbol{\mathcal{I}}_{-i}$ represents all other attributes except $\boldsymbol{\mathcal{I}}_i$. The terms then sum in the log joint distribution so that

$$\log p\left(\boldsymbol{\mathcal{I}} \mid \mathbf{l}\right) = \log p\left(\boldsymbol{\mathcal{I}}_1 \mid \mathbf{l}\right) + \log p\left(\boldsymbol{\mathcal{I}}_2 \mid \mathbf{l}\right) +, \ldots, + \log p\left(\boldsymbol{\mathcal{I}}_n \mid \mathbf{l}\right). \tag{6}$$

This provides a way to fuse multiple image attributes and prior target shape information. It allows the use of multiple images when the shape of the target in the two images can be described by a two–dimensional transformation. For example let $\boldsymbol{\mathcal{I}} = (\boldsymbol{\mathcal{I}}_1, \boldsymbol{\mathcal{I}}_2)$ be a pair of images and let $\mathbf{T}_{1,2}$ be a transformation that projects points from $\boldsymbol{\mathcal{I}}_1$ into $\boldsymbol{\mathcal{I}}_2$. Then the region likelihood for two images is

$$\log \hat{L}_r\left(\boldsymbol{\mathcal{I}} \mid v\right) = \oint_{\boldsymbol{\Gamma}} \left\{ \int_0^y \log \frac{p\left(\boldsymbol{\mathcal{I}}_1 \mid \alpha_{\boldsymbol{\alpha}}\right)}{p\left(\boldsymbol{\mathcal{I}}_1 \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right)} dy \right\} dx + \oint_{\mathbf{T}_{1,2}(\boldsymbol{\Gamma})} \left\{ \int_0^y \log \frac{p\left(\boldsymbol{\mathcal{I}}_2 \mid \alpha_{\boldsymbol{\alpha}}\right)}{p\left(\boldsymbol{\mathcal{I}}_2 \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right)} dy \right\} dx. \tag{7}$$

The parameters of the transformation $\mathbf{T}_{1,2}$ are either known or become part of the estimation process (see Section 3). The second part of the likelihood function consists of an edge term so that the template is aligned with the edges in the image. The edge likelihood using the reasoning above is

$$\log L_e\left(\boldsymbol{\mathcal{I}} \mid v\right) = \oint_{\boldsymbol{\Gamma}} \log \frac{p\left(\left|\nabla \boldsymbol{\mathcal{I}}\right| \mid \alpha_e\right)}{p\left(\left|\nabla \boldsymbol{\mathcal{I}}\right| \mid \alpha_{\bar{e}}\right)} ds, \tag{8}$$

where $|\nabla \boldsymbol{\mathcal{I}}|$ is the edge map in which edge points are taken to follow the distribution α_e and s is the arc length parameter.[8] Now that the probabilistic model has been defined for the template, the next step is to consider estimating the template parameters for a given image. This is considered next.

[†] Chakraborty *et al.*[6] first used Green's theorem to integrate region information. Their form however uses two integrations and partial boundary derivatives and they integrate over a region classified image rather than using a statistical test between region hypotheses.

The prior and the likelihood define the posterior density of the template according to Bayes theorem. Inferencing involves estimating the parameters that maximise the posterior probability for a given image. These parameters include the shape, the position and the object class of the template. Inferencing from complex distributions usually involves simulation techniques such as the Gibbs sampler[9] or the Metropolis-Hastings[10] algorithm. In these algorithms it is only necessary to have a relative probability for each possible state of the system. The posterior density of the template provides this,

$$\pi\left(v \mid \mathcal{I}\right) \propto L\left(\mathcal{I}|v\right) p\left(v\right). \tag{9}$$

The negative log of this is referred to as the energy. A transition kernel is chosen that defines the probability of the system moving to a new state given the previous state. A sequence of samples from this kernel defines a Markov chain. The choice of kernel can be such that the samples from the Markov chain approximate samples from the posterior density.[10] In Metropolis-Hastings sampling the number of estimated parameters is fixed. The number of co-ordinates needed to give sufficient resolution to match the template to an object is not known a priori however. So the number of parameters that need to be inferred is itself an unknown parameter. Also each different object can have a different prior and likelihood and therefore a different posterior density. It is necessary then that the system should be able to decide among multiple target hypotheses. Finally the parameters need to be able to undergo both discontinuous and smooth continuous changes so that the template does not converge at the first local minimum that is encountered and so that it can accommodate a wide range of shapes. In the next section an inferencing mechanism based on a jump-diffusion process is proposed which allows the template this type of behaviour.

Grenander and Miller first introduced a jump-diffusion[11] process to infer the number, shape and position of structures in mitochondria images. The individual templates were closed polygons with a fixed number of sides. The deformations were applied as local transformations to the tangents of a circular base template, where a prior was defined on the transformation parameters. A jump component was used to search across the discrete parameters including the addition and removal of object hypotheses while a diffusion component followed paths of mean steepest ascent in the posterior density with respect to the local transformations of each of the objects in the scene. Lanterman[12,13] also used a jump-diffusion process with fixed three–dimensional models to interpret forward looking infrared scenes. A method was proposed to discover the required number of objects in the image and the model alignment and position parameters necessary to describe the scene. Green[14] later developed an alternative jump algorithm that implements a Metropolis-Hastings type algorithm where the test between states of different dimensions is normalised with a Jacobian term. A variable resolution deformable template based on this algorithm has been developed by Pievatolo *et al.*[15] Rue and Hurn[16] used the same jump algorithm to interpret scenes with unknown numbers of objects.

To our knowledge the templates used in all existing approaches using jump-diffusion algorithms are either fixed[17,18] or evolve under diffusion from a base template according to the posterior model.[11] The jumps are used to add, remove and change object hypotheses in multiple object scenes or together with diffusions to vary parameters in fixed models to estimate position and pose. This has been shown to be a powerful technique for objects that can be described in terms of fixed models or deformations about a typical shape. In the application described here however, while objects within each class may have much in common, their shapes generally cannot be predicted in this way. In this respect the work presented here is similar to Storvik,[19] Peivatolo[15] and Staib *et al.*[20] who consider templates to model unknown shapes. A different probability model and inferencing process is used here however. Jumps and diffusions in our approach carry out the deformations. Together these allow a sufficiently wide range of object shapes to be modelled. The jumps permit discontinuous changes in shape, position and target model while the diffusions permit smooth continuous flowing behaviour.

Jump-diffusion is a Markov chain process although unlike Metropolis-Hastings algorithms a diffusion component operates while the process is within each fixed subspace and a jump component allows transitions between different subspaces. The frequency of jumps depends on the states reachable from the current configuration. A jump transition intensity defines when and where the jumps occur. A brief description of our use of jump-diffusion is given here though the interested reader is referred to Grenander[11] and Amit[21] for theoretical underpinnings. In the jump-diffusion algorithm proposed here each fixed subspace is a polygon with a fixed number of vertices and an associated target type. The inference process involves sampling a section of the template boundary according to the local shape function so that the least well fitting parts of the boundary are selected most often. A marginal intensity function is generated for the set of possible jumps from the current state of the template. The deformations are always restricted to the selected part of the boundary. A jump from the current state is proposed when the intensity reaches a limit that is defined by a sequence of exponential random variables. A jump proposal is sampled from a transition density that is defined by the intensity function. The proposal is either accepted or rejected depending on the change in probability. At times between jumps the template satisfies a diffusion equation where the continuous template parameters follow paths of mean steepest ascent in the posterior density.

The jump set defines five possible jump types, which are namely vertex addition, vertex deletion, vertex deformation, a template move, and a change of target type. It is written

$$\mathcal{J}\left(v\right) = \bigcup_{i=1}^{5} \mathcal{J}_i\left(v\right), \tag{10}$$

and this defines all possible jumps that the template can take a given time. The jump intensity function represents the sum of the probabilities of all accessible states that the template can reach by making a jump. It indicates at which times a jump from the current state should be proposed, and through a transition kernel it also defines which jump should be taken. The (joint) jump intensity function is

$$q\left(v\right) = \int_{v' \in \mathcal{J}(v)} q\left(v, dv'\right). \tag{11}$$

The term $q\left(v, dv'\right)$ is the (marginal) jump intensity and defines the probability of the template jumping from the current state v to a new state v'. The jump intensity used here is

$$q\left(v, dv'\right) = p_j \min\left(1, \frac{\pi\left(v' \mid \mathcal{I}\right)}{\pi\left(v \mid \mathcal{I}\right)}\right) d\left(v'\right), \tag{12}$$

where p_j is the constant prior probability of the given jump type occurring[‡]. This can be shown to satisfy Grenander's balance condition for the given jump set and intensity function. A jump is proposed at simulation time t_s such that

$$t_s = \inf\left\{t : \int_{t_{s-1}}^{t} q\left(v\right) dt > \tau_s\right\}, \tag{13}$$

where τ_s is an exponential random variable with mean defining the jump frequency, and t_{s-1} is the time of the last jump. A jump is proposed as soon as the integrated intensity exceeds the random variable τ_s. The jumps therefore occur more frequently when there are high probability states that can be reached. This formulation is based on Grenander's Gibbs jump dynamics. This provides strong inferencing capabilities, but unlike the simpler Metropolis jump dynamics where new states are taken from the prior, Gibbs jump dynamics require integrating the intensity function over parts of the jump set.

The jump transition kernel $Q\left(v, dv'\right) = q\left(v, dv'\right)/q\left(v\right)$ gives the transition probabilities for all states in the jump set $\mathcal{J}\left(v\right)$. The probability mass is concentrated at those jump points that provide the biggest energy decrease. The transition kernel is impossible to generate and sample in practice though because there are an infinite number of possible jumps. This problem is addressed here by generating a marginal jump transition kernel

$$Q_t\left(v, \mathcal{J}_i\left(v\right)\right) = \int_{t_{s-1}}^{t} \int_{v' \in \mathcal{J}_i(v)} q\left(v, dv'\right) dt, \qquad \text{where } \mathcal{J}\left(v\right) = \bigcup_{i} \mathcal{J}_i\left(v\right). \tag{14}$$

In this form each element represents a relative probability for a subset of jumps where the continuous parameters are integrated out. This can be sampled easily which now involves choosing a subset $\mathcal{J}_i$ instead of a jump point. Although the jump set can be divided arbitrarily, the sampling is most effective when closely related jumps are grouped together. The integration for each subset of jumps is approximated by defining simple priors on the internal parameters. The estimate of the transition density is then incrementally improved at each iteration by computing the intensities for a fixed number of jump points in each subset by sampling these internal parameters. At any instant t

[‡] The notation $d\left(v'\right)$ is used to represent a measurable element of space at the new state.

the sum of the transition density over all jump subsets then also defines the jump times through Equation 13. The accuracy of the marginal transition kernel depends directly on the time between jumps. In this way the transition kernel is not computed to an unnecessarily high accuracy when a particular jump is imminent. In practice it has been found that a small number of computed jump points gives sufficient accuracy. Once the jump subset $\mathcal{J}_i$ has been selected, an internal jump point is sampled. This is easy since this only involves sampling the internal parameters which are taken as either uniform or normal random variables. In this case these are either for a template move, a deformation, changing the number of vertices, or choosing a new target type.

At times between jumps, the template evolves with a fixed target type and a fixed number of vertices according to the stochastic differential equation

$$dv_{\mathbf{1},k} = \frac{1}{2}\nabla\left\{\log L\left(\boldsymbol{\mathcal{I}} \mid v\right) + \log p\left(v\right)\right\} dt + \sqrt{2}\, d\mathbf{W}_{2k}\left(t\right), \tag{15}$$

where the deterministic gradient term is the log posterior or mean forward drift potential and $d\mathbf{W}$ is the Wiener process in $2k$ dimensions. Grenander *et al.* show that the drift vector for the parameters of their templates can be expressed very concisely as a curvilinear integral (Theorem 3). In our case however, the drifts represent forces on the template boundary co-ordinates themselves because of the direct contour representation. If the energy of the template can be expressed as a functional of the boundary and its derivatives, then the variational force acting at each point on the template can be derived from the resulting Euler-Lagrange equations. The deterministic component of the diffusion equation in our case is the same as an active contour[2] where the log prior is analogous to the internal energy and the log likelihood is analogous to the image energy. The choice of probability model can easily give a drift vector that is equivalent to snakes. In this stochastic equation the process will follow mean paths of steepest ascent because of the random component which is approximated here with a vector of $2k$ independent normal variables.

Zhu *et al.*[22] recently developed a segmentation theory called region competition that unified many aspects of active contours and region growing algorithms. Starting from a global minimum description length (MDL) criteria on the segmented regions they derived a force term for each point $\vec{v}$ on a boundary,

$$\frac{d\vec{v}}{dt} = \left\{-\mu\kappa(\vec{v}) + \log\frac{p\left(\boldsymbol{\mathcal{I}} \mid \alpha_1\right)}{p\left(\boldsymbol{\mathcal{I}} \mid \alpha_2\right)}\right\}\mathbf{n}, \tag{16}$$

where $\mathbf{n}$ is the outward pointing normal and $\kappa(\vec{v})$ is the curvature. The first term is equivalent to the geometric heat equation[23] while the second term is a statistical force that depends on the Bayes factor between competing region hypotheses. The statistical force term in region competition that was derived from an MDL criterion is equal to the (functional) gradient of our region likelihood term (Equation 4). That is

$$\nabla\log L_r\left(\boldsymbol{\mathcal{I}} \mid v\right) = \oint_{\mathbf{\Gamma}} \log\frac{p\left(\boldsymbol{\mathcal{I}} \mid \alpha_1\right)}{p\left(\boldsymbol{\mathcal{I}} \mid \alpha_2\right)}\mathbf{n}\, ds. \tag{17}$$

This is the drift force component due to the region likelihood in the diffusion equation (Equation 15). Zhu found that this force was sensitive to outliers and that this could have a significant effect in noisy images. To solve this problem they averaged the value of the statistical force over a window with size dependent on the signal to noise ratio of the image. In effect this is equivalent to convolving the log term $\log p\left(\boldsymbol{\mathcal{I}} \mid \alpha_1\right)/p\left(\boldsymbol{\mathcal{I}} \mid \alpha_2\right)$ with an averaging filter to generate a smoothed force field. The method we propose is not as sensitive however because it uses both the likelihood energy in the jump process and the gradient of the likelihood at the template boundary in the diffusion process. The term $\nabla\log L_r\left(\boldsymbol{\mathcal{I}} \mid v\right)$ can be considered as a function of the contour v and the target and background probability distributions. Then for any two given distributions, α_1 and α_2, the zero crossings of the log term define a set of contours v_i in the image where the statistical force in Equation 17 is in equilibrium at every point.

In the first experiments the template was trained with three target classes containing approximately one hundred and eighty silhouettes of buildings, aircraft and clutter obtained from both real images and accurate object models.

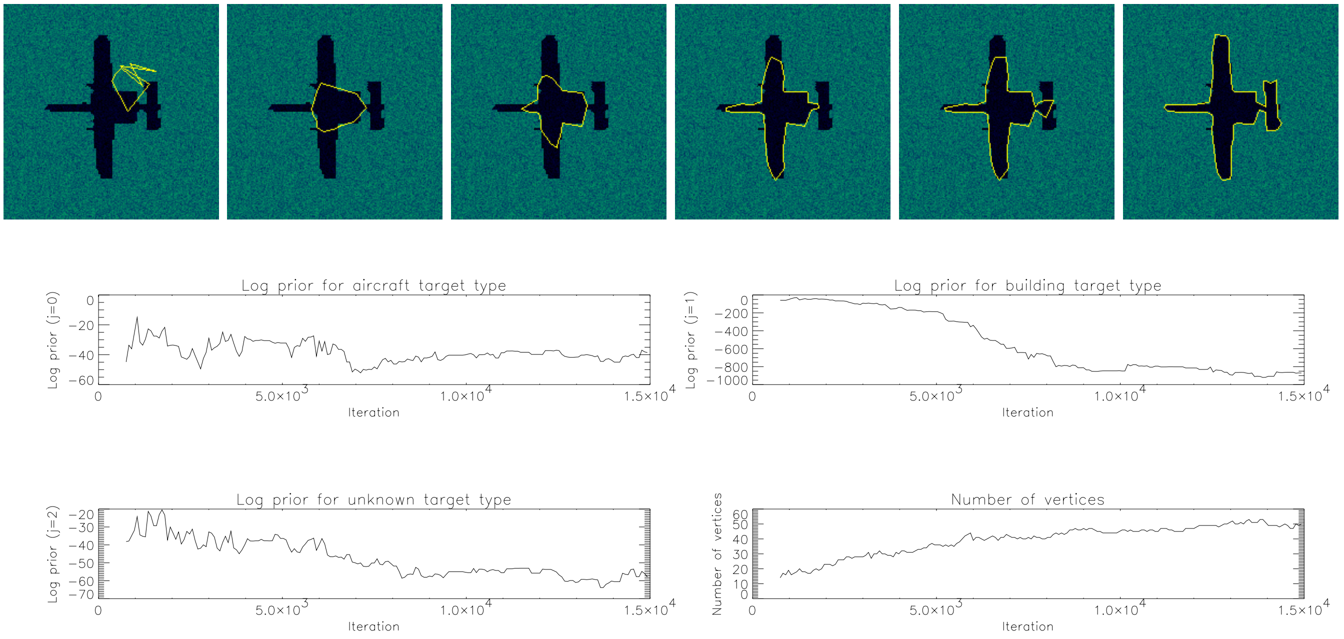
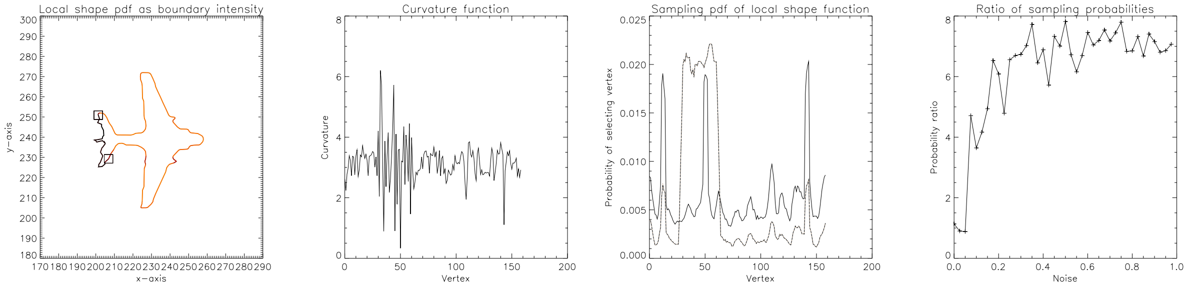


Figure 1. Extracting a synthetic shape. The template starts in an initial configuration with several intermediate states shown up to the final state. The log priors are shown for the classes together with the number of vertices at each stage.



(a) (b) (c) (d)

Figure 2. The shape in 2(a) was corrupted with Gaussian noise between the markers. Figure 2(b) shows the curvature while 2(c) shows the sampled local shape function before (solid) and after (dashed) the noise was added. Figure 2(d) shows the sampling ratio for corrupted to uncorrupted parts of the boundary.

The global shape vector contained six Fourier descriptors, the compactness and an entropy term,[24] while the local shape density was based on sequences of four vertices. The region and edge models were taken to follow Gaussian distributions with known mean and variance. The results in Figure 1 show the template fitting process. This noise degraded image consists of a shape taken from the aircraft training set. The system was trained on the remaining shapes (this was the only example of a plane of this type). The template starts in a random configuration and the sequence of frames show the evolution of the template up to the final result. The last frame shows the best fitting template according to the maximum posterior probability. The plots show the prior probabilities (after an initial burn in period) and the number of vertices. Figure 2 shows the boundary sampling method using the local shape function. The shape in Figure 2(a) was taken from the training set and corrupted with Gaussian noise between the markers. The boundary is displayed so that grey level reflects the value of the local shape function with dark areas indicating a low probability. Figure 2(b) shows the resulting curvature function while Figure 2(c) shows the density generated by sampling sections of the template boundary according to the negative log of the local shape probability. This is shown before (solid) and after (dotted) the boundary was corrupted. The height indicates the frequency at which each part of the boundary would be selected when the template is in this state. The sharp rise in the sampling frequency at vertices in the corrupted regions can be seen. Figure 2(d) shows the ratio of mean sampling frequencies.

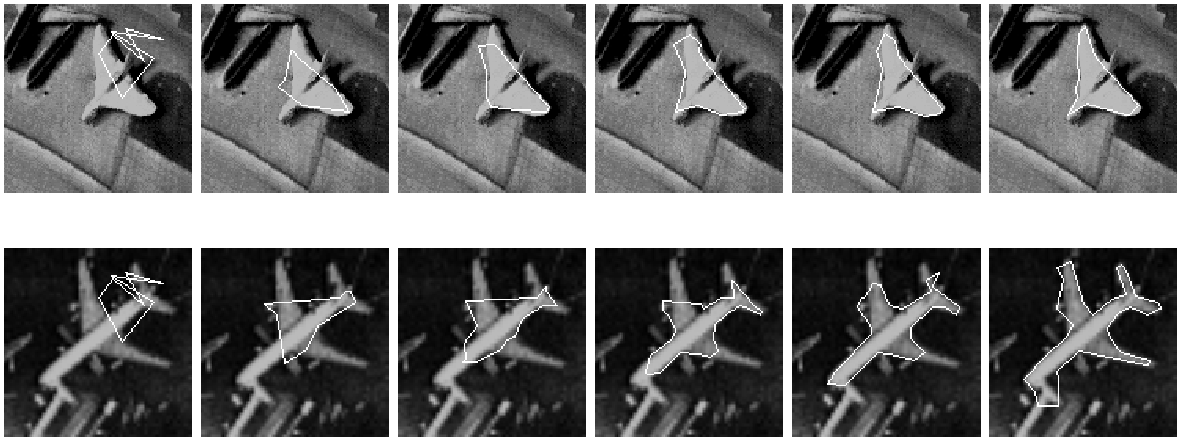


Figure 3. The evolution of the template using jump-diffusion in an infrared image (top) and an optical image.

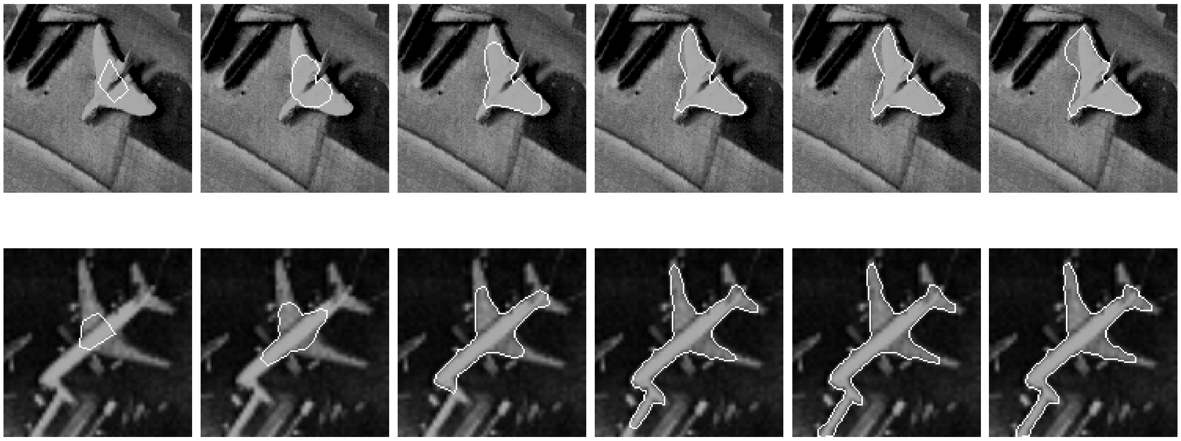


Figure 4. A comparison to show the evolution of an active contour on the same images as Figure 3. The contour is driven by a statistical region force (Equation 16) and is allowed a variable number of vertices.

The images in Figures 3 and 4 compare the jump-diffusion template with an active contour that is driven by a curvature and a region force (Equation 16). The contour is allowed to add and remove vertices as it stretches or shrinks so that it is able to represent arbitrary shapes. The jump-diffusion template in the infrared image in Figure 3 initially extends into the low contrast region by the wing in the top of the image. This part is pulled back however by the shape constraints and the likelihood which uses energy information across the whole object rather than just the energy gradient at the boundary. This does not happen in the case of the active contour in Figure 4. The contour flows freely into the low contrast region and correction is not possible since movement is deterministic and always in a single direction[§]. Similarly the shadow on the tail of the aircraft causes some distortion to the

[§] This should not be confused with the region competition algorithm.

contour. The optical image in Figure 4 highlights the problems caused by local minima. The shadow cast on the tail of the aircraft prevents the contour flowing into this region. This is because the smoothing term also prevents the contour from squeezing through narrow gaps. The jump-diffusion template however in the fourth and fifth frames in Figure 3 makes a jump over this local energy minimum to reach a better global state. The constraints on the shape also prevent the template flowing completely into the attachment at the front of the plane. In Figure 5 the results are shown for three further images with the active contour in the left two columns. In the first example the active contour correctly represents the front of the plane but fails to extract one of the wings. In the second example the low contrast regions and presumably local minima prevent a good representation of the house image while in the third example both methods give comparable results.

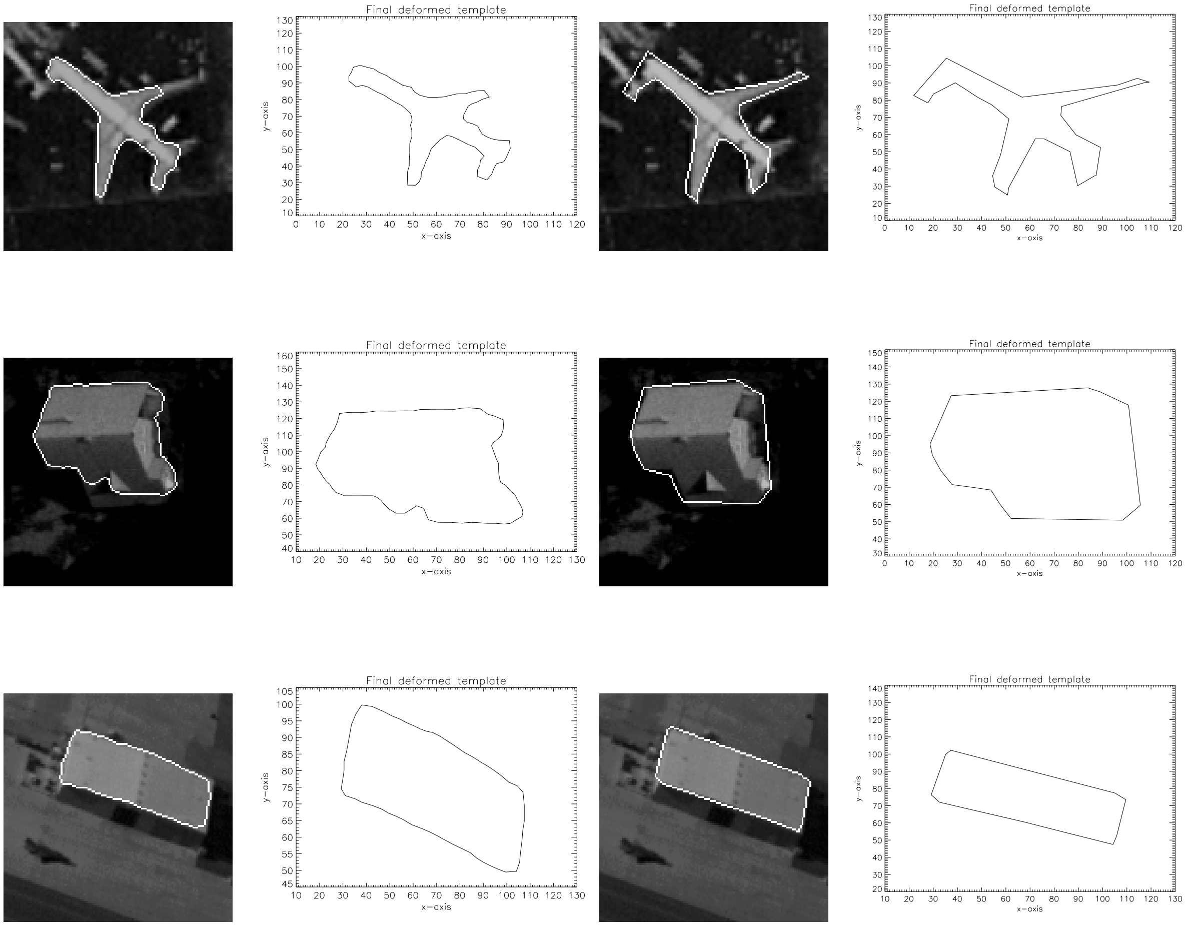


Figure 5. Comparison with a region based active contour. The two columns on the left show object extraction with an active contour. The two columns on the right show the results on the same images using jump-diffusion inferencing.

In Figure 6 the extraction of some buildings and planes are shown. Again the template started in a random state. Although these objects could have been extracted much more accurately using specialised models, the important point to note is that they were extracted using only very general models. Also, unlike region based or active contour methods the boundaries are well defined and quite realistic. The high curvature points such as building corners and the sections of the aircraft have not been smoothed and the template has not flowed too much beyond the object boundaries. Figure 7 shows target classification. The template starts in a random state and is initially in the unknown class. A jump to the correct class occurs as the template fits to the building in the image. It can be seen from the graph that the template stays in the correct class for the rest of the simulation. The target classification can either be taken as the class that the template was in at the maximum posterior state or it can be taken as the mode of the class for the simulation. The mode should be more accurate because it is based on many configurations of the template rather than a single configuration at a point in time. The processing time for the algorithm depends on the size and complexity of the shape and the accuracy at which the transition kernel is calculated. The plane images in Figure 6 needed fifteen thousand iterations while the building images needed less[¶].

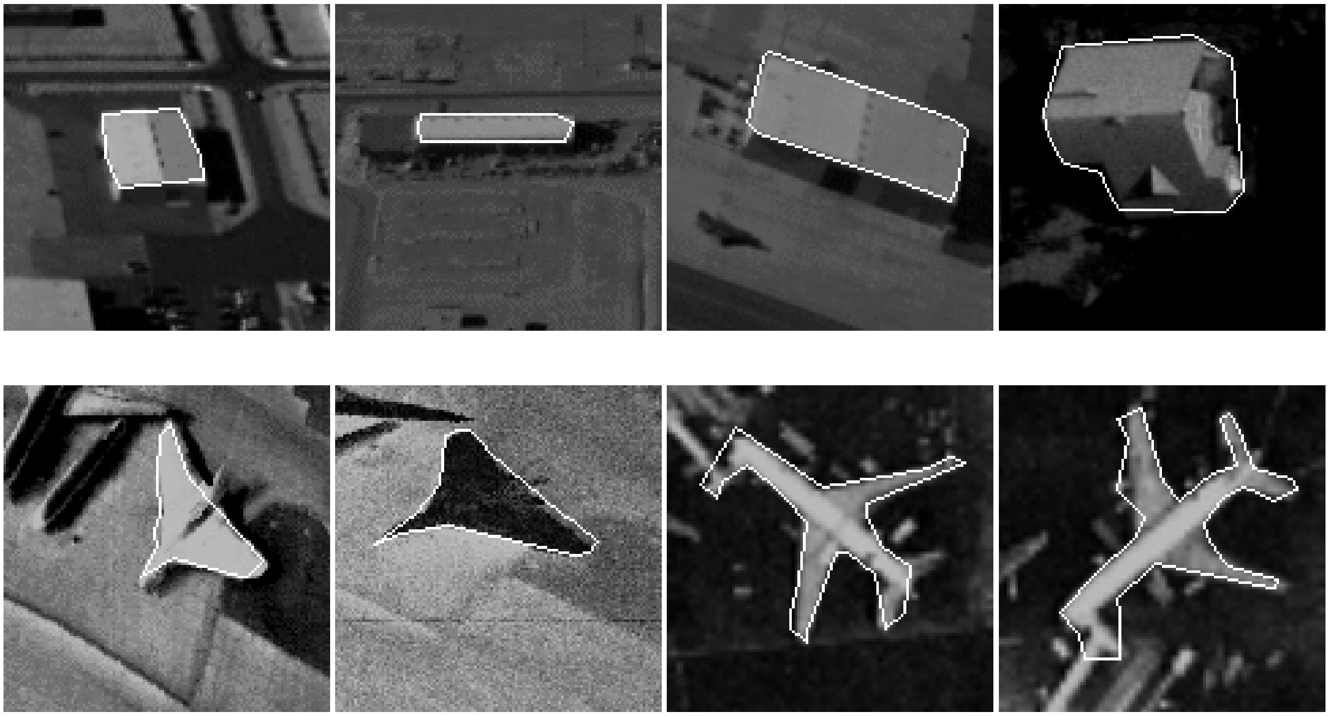


Figure 6. Extracting objects in visible and infrared images using jump-diffusion. The top row shows buildings detected in optical images. The lower row shows planes detected in both infrared and optical images.

This work has described a new type of deformable model based on a jump-diffusion process. It can be used to segment and classify two–dimensional objects when only vague prior knowledge about their shapes exists. This differs from existing methods because the template has two different types of behaviour. Jumps allow the template to undergo discontinuous changes in shape and position and to decide among multiple target models, while diffusions allow the template to undergo smooth flowing deformations. A prior was introduced that defined a measure of the template in terms of local and global shape properties. An efficient form for the image likelihood was given that extends to multiple attributes and multiple images. The template uses the energy defined by these measures in the jump process and uses their energy gradients in the diffusion process. It is less sensitive than snake methods that use only gradient information and it is less reliant on favourable initial conditions. It was shown that the jumps allow the template to avoid local minima by jumping energy barriers while diffusion provides localised refinement. A marginal jump transition density was described in which a probability for subsets of permissible jumps is computed by integrating over the continuous parameters. This is incrementally integrated over time so that its sum at any instant specifies the jump times. This allows efficient computation while providing effective inferencing. The relationship to active contours and region competition was discussed and a comparison was given. The proposed method is easily extended to allow fusion of probabilistic output from other systems. In future work we hope to investigate the fusion and recognition aspects of the system more, and to investigate other prior shape models.

[¶] Fifteen thousand iterations take between thirty and eighty seconds on a Pentium 300 with our software which is not optimal.

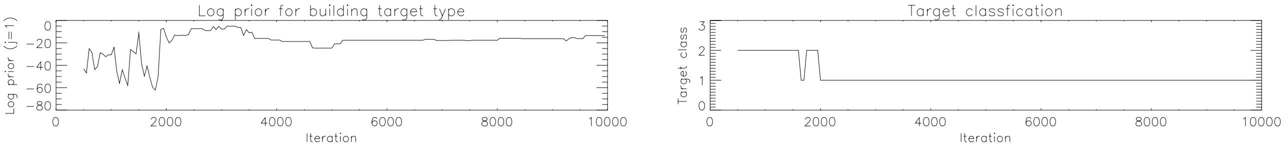


Figure 7. Classifying a building target in an infrared image. The template starts in a random configuration in the unknown class ($\mathbf{l} = 2$) and makes two jumps to the building class ($\mathbf{l} = 1$) as the template is fitted to the image. It stays in the correct class for the rest of the simulation.

Let $\mathcal{R}$ be the region enclosed by the template and let $\overline{\mathcal{R}}$ be the complement of $\mathcal{R}$, and let $\mathbf{\Gamma}$ be the boundary of the template. The joint probability for the template across a region can then be written in log form

$$\log L_r\left(\mathcal{I} \mid v\right) = \int\!\!\int_{\overline{\mathcal{R}}+\mathcal{R}} \log p\left(\mathcal{I} \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right) dx\, dy - \int\!\!\int_{\mathcal{R}} \log p\left(\mathcal{I} \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right) dx\, dy + \int\!\!\int_{\mathcal{R}} \log p\left(\mathcal{I} \mid \alpha_{\boldsymbol{\alpha}}\right) dx\, dy. \tag{18}$$

The region $\overline{\mathcal{R}} + \mathcal{R}$ can be taken as fixed, the whole image for example. The first term is then constant so it can be ignored in the log likelihood since constants will cancel in the Metropolis ratio (Equation 12). This leaves a difference between two terms based on the region enclosed by the template $\mathcal{R}$.

$$\log L_r\left(\mathcal{I} \mid v\right) = \int\!\!\int_{\mathcal{R}} \log p\left(\mathcal{I} \mid \alpha_{\boldsymbol{\alpha}}\right) dx\, dy - \int\!\!\int_{\mathcal{R}} \log p\left(\mathcal{I} \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right) dx\, dy. \tag{19}$$

Now Green's theorem can be used to convert this into a boundary integral. Let $P\left(x,y\right)$ and $Q\left(x,y\right)$ be any continuous differentiable functions. Then Green's theorem can be written[22]

$$\int\!\!\int_{\mathcal{R}} \frac{\partial P}{\partial y} - \frac{\partial Q}{\partial x} dx\, dy = \int_{\mathbf{\Gamma}} P dx + \int_{\mathbf{\Gamma}} Q dy. \tag{20}$$

The functions can be defined

$$P\left(x,y\right) = \int_0^y \log p\left(\mathcal{I} \mid \alpha_i\right) dy \qquad \text{and} \qquad Q\left(x,y\right) = 0, \tag{21}$$

so that Green's theorem gives

$$\int\!\!\int_{\mathcal{R}} \log p\left(\mathcal{I} \mid \alpha_i\right) dx\, dy = \oint_{\mathbf{\Gamma}} \left\{ \int_0^y \log p\left(\mathcal{I} \mid \alpha_i\right) dy \right\} dx. \tag{22}$$

Then using Equation 22 it is possible to convert the likelihood in Equation 19 into the boundary integral

$$\begin{aligned} \log L_r\left(\mathcal{I} \mid v\right) &= \oint_{\mathbf{\Gamma}} \left\{ \int_0^y \log p\left(\mathcal{I} \mid \alpha_{\boldsymbol{\alpha}}\right) dy - \int_0^y \log p\left(\mathcal{I} \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right) dy \right\} dx, \\ &= \oint_{\mathbf{\Gamma}} \left\{ \int_0^y \log \frac{p\left(\mathcal{I} \mid \alpha_{\boldsymbol{\alpha}}\right)}{p\left(\mathcal{I} \mid \alpha_{\overline{\boldsymbol{\alpha}}}\right)} dy \right\} dx. \end{aligned} \tag{23}$$

The part in the brackets is dependent only on the image and x and y. It can therefore be computed independently of the template and stored as a constant probability integral image so that the likelihood can be computed very quickly. The form of the probability density is general and could be a non parametric function, for example something generated by another target detection system.

Thanks to Tim Field, Dave Hutber and Richard Glendinning for reviewing.

1. D. Booth, A. Reno, S. Foulkes, P. Kent, K. Hermiston, and S. Lewis, "Automatic interpretation of IR and optical reconnaissance imagery," *Proceedings of the SPIE: Signal Processing, Sensor Fusion, and Target Recognition VI* **3068**, pp. 158–169, 1997.
2. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision* **1**(4), pp. 321–331, 1988.
3. U. Grenander, Y. Chow, and D. Keenan, *Hands: A pattern-theoretic study of biological shape*, Springer-Verlag, 1991.
4. R. Brooks, "Model based three-dimensional interpretation of two-dimensional images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **5**(2), pp. 140–150, 1983.
5. A. L. Reno, "Detecting buildings in aerial images using shape descriptors," *Proceedings of the Sixth International Conference on Image Processing and its Applications* **2**, pp. 468–472, 1997.
6. A. Chakraborty, L. Staib, and J. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Transactions on Medical Imaging* **15**(6), pp. 859–870, 1996.
7. P. Lee, *Bayesian statistics: An introduction*, Edward Arnold, 1989.
8. M. do Carmo, *Differential geometry of curves and surfaces*, Prentice-Hall Inc., 1976.
9. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(6), pp. 721–741, 1984.
10. W. Hastings, "Monte Carlo sampling methods using Markov chains, and their applications," *Biometrika* **57**(2), pp. 97–109, 1970.
11. U. Grenander and M. Miller, "Representations of knowledge in complex systems (with discussion)," *Journal of the Royal Statistical Society B* **56**(4), pp. 459–603, 1994.
12. A. Lanterman, "Jump-diffusion algorithms for the automated understanding of forward-looking infrared scenes," Master's thesis, Washington University, St Louis, Missouri, 1995.
13. A. Lanterman, M. Miller, and D. Snyder, "Implementation of jump-diffusion algorithms for understanding FLIR scenes," *Proceedings of the SPIE: Automatic Object Recognition V* **2485**, pp. 309–320, 1995.
14. P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrica* **82**(4), pp. 711–732, 1995.
15. A. Pievatolo and P. Green, "Object restoration through dynamic polygons," tech. rep., University of Bristol, 1995.
16. H. Rue and M. Hurn, "Bayesian object identification," tech. rep., Norwegian Institute of Science and Technology, Trondheim, Norway, 1997.
17. A. Lanterman, M. Miller, and D. Snyder, "Representations of thermodynamic variability in the automated understanding of FLIR scenes," *Proceedings of the SPIE: Automatic Object Recognition* **2756**, pp. 26–37, 1996.
18. D. Phillips and A. Smith, "Bayesian model comparison via jump-diffusions," in *Markov chain Monte Carlo in practice*, W. Gilks, S. Richardson, and D. Spiegelhalter, eds., ch. 13, pp. 215–239, Chapman & Hall, 1996.
19. G. Storvik, "A Bayesian approach to dynamic contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(10), pp. 976–986, 1994.
20. L. Staib and J. Duncan, "Boundary finding with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(11), pp. 1061–1075, 1992.
21. Y. Amit and M. Miller, "Ergodic properties of jump-diffusion processes," tech. rep., Department of Statistics, University of Chicago, 1997.

22. S. Zhu and A. Yuille, "Region competition - unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(9), pp. 884–900, 1996.
23. M. Gage and R. Hamilton, "The heat equation shrinking convex plane curves," *Journal of Differential Geometry* **23**, pp. 69–96, 1986.
24. L. Oddo, "Global shape entropy: A mathematically tractable approach to building extraction in aerial imagery," *Proceedings of the SPIE: Computer Vision Applications* **1623**, pp. 91–101, 1992.