In *Maximum Entropy and Bayesian Methods.*, K.M. Hanson and R.N. Silver, eds., pp. 125-134, Kluwer Academic, Dordrecht, 1996.

# THE BAYES INFERENCE ENGINE

K.M. HANSON AND G.S. CUNNINGHAM
*Los Alamos National Laboratory, MS P940*
*Los Alamos, New Mexico 87545, USA*†

**Abstract.** We are developing a computer application, called the Bayes Inference Engine, to provide the means to make inferences about models of physical reality within a Bayesian framework. The construction of complex nonlinear models is achieved by a fully object-oriented design. The models are represented by a data-flow diagram that may be manipulated by the analyst through a graphical-programming environment. Maximum *a posteriori* solutions are achieved using a general, gradient-based optimization algorithm. The application incorporates a new technique of estimating and visualizing the uncertainties in specific aspects of the model.

**Key words:** Bayesian analysis, MAP estimator, uncertainty estimation, object-oriented programming, adjoint differentiation, optimization

## 1. Introduction

As scientists, we use models to describe and understand physical reality. In building our models from data, we must be able to answer such questions as: Which models are appropriate? What are the values of the model parameters? How certain can we be in our interpretations drawn from measurements that are subject to uncertainty? The methodology of Bayesian analysis provides the framework to address these questions. In the Bayesian approach our degree of certainty is represented as a probability density function. Appropriate calculation and use of the posterior, the probablity that summarizes all available information concerning a particular physical situation, permits us to quantitatively answer the above questions regarding our models of the physical world.

We are developing a computer application, which we call the Bayes Inference Engine (BIE), to provide a tool with which to easily perform Bayesian analysis of physical models. The BIE represents a computational approach to Bayesian inference, as opposed to the traditional analytical approach [1]. The computational approach affords great flexibility in modeling, which facilitates the construction of complex models for the objects under study. For instance, the BIE easily deals

†Email: kmh@lanl.gov, cunning@lanl.gov

with data that are nonlinearly dependent on the model parameters. For example, radiographic data are not linearly related to material densities [2]. One of our first goals is to reconstruct objects from several radiographs taken of them. Furthermore, the computational approach allows one to use nonGaussian probability distributions, such as entropic priors, which have been used with great success in certain kinds of applications [3,4].

Given the context of these Proceedings, it should not be necessary to provide a detailed description of Bayesian analysis. To learn the basics of Bayesian analysis, the reader is referred to the collected works of the Proceedings of this workshop series, in particular to those by Gull and Skilling and their colleagues [3–5], to textbooks that explain the modern view of Bayesian methodology [6,7], or to several introductory articles related to image analysis written by one of the authors [8,9]. Lack of space also precludes us from showing examples produced by the BIE. The reader is encouraged to look at some of the many references cited, many of which are available on the World Wide Web (`http://home.lanl.gov/kmh`).

## 2. The Bayes Inference Engine

Our goals for the BIE are that it should be easy to learn and to use and that it should provide a high degree of interactivity with good visualization of the inference process and the models. Additionally, we are building an application that provides the user with a great deal of flexibility in configuring object models and measurement models. We deem these features essential to the usefulness of the BIE.

In Bayesian analysis, the state of knowledge about the parameters $\mathbf{x}$ associated with a model that describes the physical object being studied is summarized by the posterior, which is the probability density function $p(\mathbf{x}|\mathbf{d})$ of the parameters given the observed data $\mathbf{d}$. Bayes law gives the posterior as

$$p(\mathbf{x}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{x})\, p(\mathbf{x}) \quad . \tag{1}$$

The probability $p(\mathbf{d}|\mathbf{x})$, called the likelihood, comes from a comparison of the actual data to the data predicted on the basis of the model of the object. The predicted data are generated using a model for how the measurements are related to the object, which we call the measurement model. The prior $p(\mathbf{x})$ expresses what is known about the object, exclusive of the present measurements, and may represent knowledge acquired from previous measurements, specific information regarding the object itself, or simply general knowledge about the parameters, e.g. that they are nonnegative. Bayes law says that for a given object model the posterior can be evaluated by combining the likelihood, which requires the data values predicted for that object model, and with the numerical value of the prior. This calculation is usually straightforward. It involves calculating the predicted measurements for the given object model, which we refer to as the forward measurement calculation.

A typical nonBayesian approach to estimating model parameters from a given set of data is to attempt to apply the inverse of the forward measurement process to the data. Such an approach is plagued by problems, particularly when there are insufficient data to uniquely determine all aspects of the actual object or when the measurements are degraded by excessive noise. Common remedies for overcoming such problems is to invoke some sort of regularization to permit the inverse solution or to reduce in the model the number of parameters to be determined.

We avoid the calculational difficulties of direct inversion in the BIE by basing the estimation procedure on the forward measurement calculation, which results in an evaluation of $\varphi = -\log[p(\mathbf{x}|\mathbf{d})]$ (neglecting a constant normalization term). The parameters for the object model are found using an algorithm to minimize $\varphi$ with respect to those parameters, which results in the well known maximum *a posteriori* (MAP) solution because it maximizes the posterior. This optimization process is facilitated by making use of the derivatives of $\varphi$ with respect to the object parameters, which are calculated using the adjoint differentiation technique described in Sect. 4. The use of priors provides the means of regularization in a probabilistic way that has a quantifiable basis, which may potentially be verified experimentally. We note that this numerical approach has many benefits, which were outlined in the Introduction. Thus with the computer we can obtain accurate Bayesian solutions to fairly complex problems that are intractable using analytic approaches. The computer also allows us to explore complex situations employing data visualization to enhance understanding, fulfilling the promise of using the full posterior provided by the Bayesian approach.

The BIE incorporates many innovative features, including:
1) a graphical programming tool programmed in an object-oriented language, which greatly enhances the flexibility of modeling objects and measurements,
2) adjoint differentiation to calculate the gradient of $\varphi$, with respect to all object parameters,
3) new approaches to solving the constrained optimization problem, which is required to find the MAP solution,
4) geometrical representations of physical objects, and
5) a new method to explore the reliability of the Bayesian solution.

We will describe each of these new developments in the following sections.

## 3. Data Flow Diagram

The analyst interacts with the BIE through a graphical programming environment [10], as shown in Fig. 1. We believe that this mode of interaction provides a very intuitive interface for building models because most scientists and engineers have had some experience with data-flow diagrams. The square icons represent transforms, which are connected by lines drawn between them to describe flow of data.

We are programming the BIE using the object-oriented (OO) language Smalltalk in the version supplied by ParcPlace Systems[1], which includes a complete class library, including classes for easy development of a graphical user-interface. In our description of object classes in the BIE, we capitalize the class names in accordance with the naming convention of this object-oriented language. A distinctive

[1]ParcPlace Systems, Inc., 999 East Arques Ave., Sunnyvale, CA 94086, tel: 408-481-9090

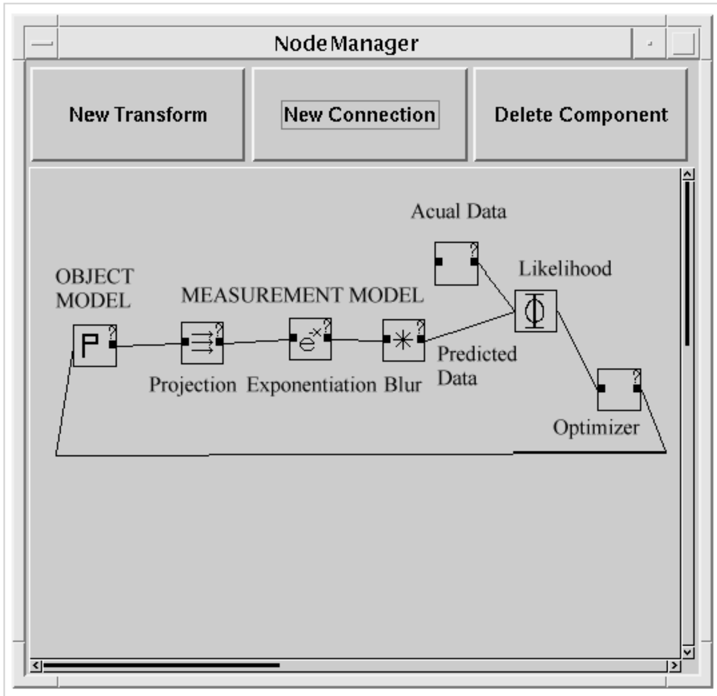


*Figure 1.* The canvas of the Bayes Inference Engine permits one to specify a data-flow diagram by connecting together Transforms, represented visually by squares on the canvas.

feature of the BIE is that the Transforms represented in Fig. 1 are 'living' objects with which one can interact. By clicking on the icon representing the Transform with the middle mouse button, a menu pops up that allows one to specify a request of the Transform. One can see a description of a Transform and change the parameters that define it. One can have the Transform display its output data structure.

Referring to the data-flow diagram in Fig. 1, the Parameters of the object model (the lefthand icon) provide input to the measurement model. The radiographic measurement model shown consists of the next three icons, which sequentially take the projection of the object, exponentiate the result, and perform a convolution with a point-spread function kernel to account for radiographic blur. The output of the measurement model represent predicted Data, which is fed into a (minus) LogLikelihood function, designated by $\Phi$, along with the actual data, the uppermost icon. A LogPrior, which operates on the model parameters, can also be specified. The output of the LogLikelihood is fed into the Optimizer, the lower right-hand icon, whose task it is to find the values of the object-model parameters that result in a minimum value for $\Phi$. One specifies the Parameters of the object model that are to be optimized by connecting their icons to the Optimizer. After optimization, the object model and its Parameter values represent the MAP

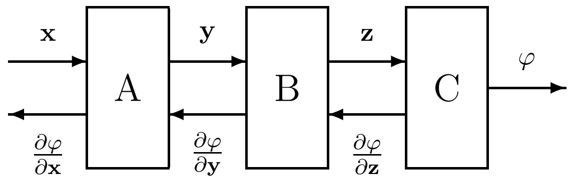


*Figure 2.* A generic data-flow diagram showing a sequence of transformations, represented by the boxes A, B, and C, starting with the data structure $\mathbf{x}$ and resulting in the scalar $\varphi$. The flow of data for the adjoint derivatives is in the reverse direction, from right to left.

solution.

We have come to realize that the OO approach has provided much more than just a productive programming environment; it has aided in the design of the overall application, as well as the numerical algorithms at a basic level. For example, the following aspects of the BIE have been elucidated by OO design: the adjoint differentiation technique, the accommodation of constraints in optimization, the automatic connection of the Optimizer to any parameter, and the appropriate role of Connectors and Transforms in the data-flow diagram. An interesting aspect of the OO design of the data-flow diagram is that no supervisor of the sequence of calculations is needed; the Transforms simply do what they are asked to do, when they are asked to do it.

## 4. Adjoint Differentiation

To obtain the MAP estimate in the BIE, we need to minimize the scalar function $\varphi$ by varying the variables that comprise the parameters of the object model. This optimization problem would be intractable without knowing the gradient of $\varphi$, or sensitivities, with respect to the many parameters on which it depends. We have uncovered a technique to calculate these crucial sensitivities, called adjoint differentiation [11], that is apparently little known. Using the adjoint differentiation technique the calculation of all these derivatives can be done in a computational time that is comparable to the forward calculation through the data-flow diagram. The adjoint sensitivity technique is crucial to the efficient operation of the BIE. We believe that it could be beneficially employed in many types of forward modeling codes. See [1] for more details.

Consider a calculation such as the sequence of transformations depicted in Fig. 2. In the context of the BIE, the transformations are implemented by the Transform class. The independent variables in the data structures designated by the vector $\mathbf{x}$ are transformed by block A to produce the dependent variables $\mathbf{y}$. This is transformed by block B to produce the dependent data structure $\mathbf{z}$, and finally by C to produce the scalar $\varphi$, which would correspond to the minus logarithm of the posterior in the BIE.

We make no assumptions about the transformations except that they are differentiable. The transforms can be nonlinear, as is the case for projections of objects defined in terms of their geometry (see Sect. 6) and for exponentiation, which is needed to model radiographic measurements.

The flow diagram indicates that $\varphi$ depends on $\mathbf{z}$, which depends on $\mathbf{y}$, which in turn depends on $\mathbf{x}$, the independent variables. The chain rule for differentiation gives the derivatives of $\varphi$ with respect to the $i$th component of $\mathbf{x}$,

$$\frac{\partial \varphi}{\partial x_i} = \sum_{jk} \frac{\partial \varphi}{\partial z_k} \frac{\partial z_k}{\partial y_j} \frac{\partial y_j}{\partial x_i} \ . \tag{2}$$

Even if the transformations are nonlinear, this expression amounts to a product of matrices, each element of which specifies the differential response of an output variable with respect to a differential change of an input variable.

The essence of adjoint differentiation is to perform the sum on the indices in the reverse order from that used in the forward calculation. If the sum on $k$ is done before that on $j$, the sequence of calculations is

$$\mathbf{I} \xrightarrow{\mathrm{C}'^{\dagger}} \frac{\partial \varphi}{\partial \mathbf{z}}^{\dagger} \xrightarrow{\mathrm{B}'^{\dagger}} \frac{\partial \varphi}{\partial \mathbf{y}}^{\dagger} \xrightarrow{\mathrm{A}'^{\dagger}} \frac{\partial \varphi}{\partial \mathbf{x}}^{\dagger} ,$$

where $\mathbf{I}$ represents the indentity vector and, for example, $\mathrm{B}'^{\dagger}$ implements the adjoint of the matrix $\frac{\partial \mathbf{z}}{\partial \mathbf{y}}$. This sequence implies intermediate data structures (e.g. $\frac{\partial \varphi}{\partial \mathbf{y}}$) that resemble the normal data structures (e.g. $\mathbf{y}$). Thus the requirement for storing these data structures is merely double that required to store the structures for the forward calculation, which may be required for the sensitivity calculation if the transformations are nonlinear. The backward flow of the adjoint derivatives is depicted in Fig. 2.

If the sums over the indices were done in the opposite order, mimicking the forward calculation, the intermediate data structures would be matrices containing a number of elements equal to the product of the sizes of the forward data structures. Since we are considering very large forward data structures, this way of accumulating the sensitivities could be untenable.

Our use of objects to represent transformations greatly helps implement this adjoint calculation [12]. In accordance with the OO approach, each transformation is self-contained; it requires only its input variables to calculate its output variables, e.g. module B uses only its input $\mathbf{y}$ to calculate its output $\mathbf{z}$. Therefore, each transformation should require nothing more than its input to implement the derivative of its output variables with respect to its input variables. We emphasize that in the OO approach we are using, each Transform has the responsibility to propagate the adjoint derivative from its output side to its input side. The Transform "knows how" to do this because it knows how to accomplish the forward calculation.

In reality, what this means is that when a new Transform is created by a programmer, the code for propogating the adjoint derivative should be developed using the logic of the forward calculation to determine the derivatives of the output variables with respect to the input variables. We stress that the derivative matrix need not be explicitly calculated and stored. The adjoint code has the responsibility of calculating the *effect* of multiplying derivatives on its output side by the derivative matrix, which can often be achieved using computer code that is very similar to the code for the forward calculation. To illustrate, for linear transformations, which can be characterized in terms of a matrix multiplication, the implementation of the adjoint differentiation calculation simply amounts to multiplication by the transpose of that matrix. In this case the transpose operation can often be realized by trivially altering the computer code for the forward transformation.

## 5. Optimization

In the BIE the MAP solution is found by minimizing $\varphi$ with respect to all the model parameters. Our optimization procedure is based on knowing the gradient of $\varphi$ with respect to the parameters, which is calculated as described in the previous section. Our use of OO programming imposes certain restrictions on how the optimization can be implemented. For example, many of the models in the BIE impose constraints on parameters. Some constraints involve fixed limits on individual parameters, e.g. nonnegativity. Constraints can also exist between parameters. The approach to optimization must include these in an OO way. That is, the Optimizer should only request that the Parameters act on themselves. Examples of possible actions include a) add a specified vector to the present values of the parameters and b) satisfy constraints on the parameters.

The general method that we employ to guarantee that the constraints on the parameters are met is by projection onto convex sets (POCS) [13]. Each Parameter checks whether constraints are violated. If they are, the Parameters minimally change themselves to meet the constraints [14].

## 6. Geometric Representation of Objects

Deformable models have been developed in a number of fields to describe objects geometrically, particularly in computer vision where the aim is to decompose a scene (image) in terms of geometrical objects [15–17].

We are pioneering the use of deformable geometric models to improve tomographic reconstructions of objects from just a few views [18–20]. This tack is quite different from the normal one of representing a 2D object in terms of its density, typically described by square pixels on an ordered grid. The reconstruction process amounts to deforming an initial object geometry in a minimal way to match the data. In the Bayesian approach, one controls the geometric deformation by placing a prior on it. The net effect is to add to $\varphi$ a deformation energy that penalizes larger deformations. This approach has proven to be a valuable means to achieve good reconstructions in situations where all other methods fail, for example when only two radiographs are available [20]. However, it must be emphasized that this approach can only be successful when the objects being reconstructed have a fairly simple morphology that is approximately known beforehand.

## 7. Exploration of the uncertainty in a Bayesian solution

One of the most important features of the Bayesian approach is that the posterior characterizes of the degree of certainty in the models used in an analysis. Although many articles have been written about Bayesian image analysis, surprisingly few of these have fully exploited the full posterior as a measure of uncertainty. The reason probably lies in the fact that it is computationally difficult to cope with the posterior in a large dimensional space.

One way to visualize the reliability of an inferred model is to display a sequence of solutions that are randomly chosen from the posterior probability distribution. This approach, proposed by Skilling et al. [21], provides a stochastic look at the range of possible solutions. The sequence of images, typically calculated off line, is presented as a video loop. By showing a representative range of alternative solutions, the degree of variability of this presentation provides the viewer with a visual impression of the degree of uncertainty in the inferred model. One would expect that the present emphasis in Bayesian research on Markov chain Monte Carlo methods [22] to generate random samples of the posterior will be useful for this type of visualization.

We have proposed a new approach [23,24], which is based on drawing an analogy between $\varphi$ and a physical potential. Then the gradient of $\varphi$ is analogous to a force. From this viewpoint an unconstrained MAP solution can be interpreted as the situation in which the forces on all the variables in the problem balance so that the net force on each variable is zero. Further, when a variable is perturbed from the MAP solution, the derivative of $\varphi$ with respect to that variable is the force that drives it back towards the MAP solution. The phrase "force of the data" takes on real meaning in this context.

To explore the reliability of a particular feature of a MAP solution, the user specifies it by directly perturbing the selected combination of parameters that characterize the feature of interest. The posterior is incremented on the basis of this perturbation to effectively apply a constant force to the parameters in question. Then, all the parameters are readjusted to minimize the new $\varphi$. The uncertainty in the parameters is indicated by the amount that they move away from their MAP values for a given applied external force. The correlations between parameters experiencing the external force and the others is demonstrated by how much and in what direction the parameters change. We have shown that this approach leads to a quantitative estimate for an appropriate part of the covariance matrix for problems in which the parameters are unconstrained [23,24]. Ideally, these correlations could be seen through direct interaction with a rapidly-responding dynamical Bayesian system. Alternatively, they may be demonstrated as a video loop produced off line.

## 8. Future Directions

The BIE provides us with an ideal basic tool with which to make Bayesian inferences regarding physical models. We aim to extend its existing capabilities in a number of directions.

In developing the BIE we have concentrated on two-dimensional models, in order to impact our programmatic goals. We expect to develop both 1D and 3D models. One-dimensional models will enable us to develop demonstrations of how the Bayesian approach addresses many familiar problems such as deconvolution, or restoration, of blurred 1D signals, spectral estimation, interpolation, and line fitting. Calculations in 1D can be done very quickly, so that the concepts discussed in Sect. 7 can be demonstrated in a real-time interactive environment. We also expect to develop three-dimensional modeling capabilities before long so that we can address the problem of tomographic reconstruction of 3D objects.

The interactivity of the BIE allows the analyst to fully diagnose the models he creates. Feedback about what is needed from the model to match the data is provided by displaying the gradient of $\varphi$, which shows the force of the data. The full interactivity with the object models makes it easy to augment the models to achieve a better fit to the data. The Bayesian methodology allows one to make inferences about the choice of models appropriate to describe reality. Our preference for simpler models over more complex ones can be incorporated through a prior on model complexity [5]. An interesting example of model selection is supplied in the context of an object defined by its boundary, which is smooth by default. The boundary might be allowed to develop a kink, i.e. an abrupt change in slope, thereby negating the smoothness constraint at a particular place, if the data provide enough evidence for such a departure from the default model [25]. We anticipate that the gradient of $\varphi$ will play a fundamental role in making such such decisions about when the complexity of a model needs to be increased.

We will implement the means to generate random samples from the posterior [22]. This capability could be used to estimate the posterior mean (as an alternative to the posterior mode) and variance, which is one way to summarize the uncertainty in solutions. This technique permits marginalization with respect to any nuisance parameters. It also can provide a visualization of the uncertainty in solutions by displaying as a video loop the sequence of random samples [21].

## Acknowledgements

This work was supported by the United States Department of Energy under contract number W-7405-ENG-36. We would like to acknowledge the helpful conversations we have had with the following people: John Skilling, Stephen F. Gull, Kyle J. Myers, and Robert F. Wagner.

## References

1. K. M. Hanson and G. S. Cunningham, “A computational approach to Bayesian inference,” in *Proc. Interface Conf.*, M. Meyer, ed., 1995 (to be published).
2. K. M. Hanson, “A Bayesian approach to nonlinear inversion: Abel inversion from x-ray data,” in Transport Theory, Invariant Imbedding, and Integral Equations, *Lect. Notes in Pure and Appl. Math.* **115**, P. Nelson, V. Fabor, D. L. Seth, and A. B. White, Jr., eds., pp. 363–368, Marcel Dekker, New York, 1989.
3. J. Skilling and R. K. Bryan, “Maximum entropy image reconstruction: general algorithm,” *Mon. Not. R. Ast. Soc.*, **211**, pp. 111–124, 1984.

4. S. F. Gull, "Developments in maximum-entropy data analysis," in *Maximum Entropy and Bayesian Methods*, J. Skilling, ed., pp. 53–71, Kluwer Academic, Dordrecht, 1989.
5. S. F. Gull, "Bayesian inductive inference and maximum entropy," in *Maximum Entropy and Bayesian Methods in Science and Engineering (Vol. 1)*, G. J. Erickson and C. R. Smith, eds., pp. 53–74, Kluwer Academic, Dordrecht, 1989.
6. S. J. Press, *Bayesian Statistics: Principles, Models, and Applications*, Wiley, New York, 1989.
7. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, Chapman & Hall, London, 1995.
8. K. M. Hanson, "Bayesian and related methods in image reconstruction from incomplete data," in *Image Recovery: Theory and Application*, H. Stark, ed., pp. 79–125, Academic, 1987.
9. K. M. Hanson, "Introduction to Bayesian image analysis," in *Image Processing*, M. H. Loew, ed., *Proc. SPIE*, **1898**, pp. 716–731, 1993.
10. G. S. Cunningham, K. M. Hanson, G. R. Jennings, Jr., and D. R. Wolf, "An object-oriented implementation of a graphical-programming system," in *Image Processing*, M. H. Loew, ed., *Proc. SPIE*, **2167**, pp. 914–923, 1994.
11. W. C. Thacker, "Automatic differentiation from an oceanographer's perspective," in *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, A. Griewank and G. F. Corliss, eds., pp. 191–201, SIAM, Philadelphia, 1991.
12. G. S. Cunningham, K. M. Hanson, G. R. Jennings, Jr., and D. R. Wolf, "An interactive tool for Bayesian inference," in *Review of Progress in Quantitative Nondestructive Evaluation*, D. O. Thompson and D. E. Chimenti, eds., vol. 14A, pp. 747–754, Plenum, New York, 1995.
13. D. C. Youla, "Mathematical theory of image restoration by the method of convex projections," in *Image Recovery: Theory and Application*, H. Stark, ed., pp. 29–77, Academic, 1987.
14. G. S. Cunningham, K. M. Hanson, G. R. Jennings, Jr., and D. R. Wolf, "An object-oriented optimization system," in *Proc. IEEE Int. Conf. Image Processing, vol. III*, pp. 826–830, IEEE, 1994.
15. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Inter. J. Comp. Vision*, **1**, pp. 321–331, 1988.
16. R. Szeliski, "Probabilistic modeling of surfaces," *Proc. SPIE*, **1570**, pp. 154–165, 1991.
17. U. Grenander and M. Miller, "Representations of knowledge in complex systems," *Jour. Roy. Stat. Soc. B*, **56**, pp. 549–603, 1994.
18. K. M. Hanson, "Flexible prior models in Bayesian image analysis," in *Maximum Entropy and Bayesian Methods*, A. Mohammad-Djafari and G. Demoment, eds., pp. 399–406, Kluwer Academic, Dordrecht, 1993.
19. K. M. Hanson, "Bayesian reconstruction based on flexible prior models," *J. Opt. Soc. Amer. A*, **10**, pp. 997–1004, 1993.
20. K. M. Hanson, G. S. Cunningham, G. R. Jennings, Jr., and D. R. Wolf, "Tomographic reconstruction based on flexible geometric models," in *Proc. IEEE Int. Conf. Image Processing, vol. II*, pp. 145–147, IEEE, 1994.
21. J. Skilling, D. R. T. Robinson, and S. F. Gull, "Probabilistic displays," in *Maximum Entropy and Bayesian Methods*, W. T. Grandy, Jr. and L. H. Shick, eds., pp. 365–368, Kluwer Academic, Dordrecht, 1991.
22. J. Besag, P. Green, D. Higdon, and K. Mengersen, "Bayesian computation and stochastic systems," *Stat. Sci*, **10**, pp. 3–66, 1995.
23. K. M. Hanson and G. S. Cunningham, "The hard truth," in *Maximum Entropy and Bayesian Methods*, J. Skilling, ed., Kluwer Academic, Dordrecht, 1994 (to be published).
24. K. M. Hanson and G. S. Cunningham, "Exploring the reliability of Bayesian reconstructions," in *Image Processing*, M. H. Loew, ed., *Proc. SPIE*, **2434**, pp. 416–423, 1995.
25. K. M. Hanson, R. L. Bilisoly, and G. S. Cunningham, "Kinky tomographic reconstruction," to be published in *Image Processing*, M. H. Loew and K. M. Hanson, eds., *Proc. SPIE*, **2710**, 1996.