

## OPERATION OF THE BAYES INFERENCE ENGINE

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

**Abstract.** We have developed a computer application, called the Bayes Inference Engine, to enable us to make inferences about models of a physical object from radiographs taken of it. In the BIE calculational models are represented by a data-flow diagram that can be manipulated by the analyst in a graphical-programming environment. We demonstrate the operation of the BIE in terms of examples of two-dimensional tomographic reconstruction including uncertainty estimation.

**Key words:** Bayesian analysis, MAP estimator, uncertainty estimation, radiographic modeling, Markov chain Monte Carlo, adjoint differentiation, object oriented design

### 1. Introduction

We developed the Bayes Inference Engine (BIE) as a modeling tool for Bayesian data analysis. While its intended application is analysis of radiographic data in terms of geometric models of the objects being radiographed, we developed it with an eye toward generality. Thus, the design approach is readily extendible to other kinds of data analysis. Significant aspects of the BIE include the great flexibility with which object models and measurement models can be configured, the high degree of interactivity with those models, and the visualization of the overall modeling process.

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. The BIE easily deals with data that are non-linearly dependent on the model parameters. For example, radiographic data are not linearly related to material densities [2]. Furthermore, the computational approach allows one to use nonGaussian probability distributions, such as likelihood functions based on Poisson distributions.

This paper is devoted to demonstrating the operation of the BIE. We will mention enough of the underlying technology to make the basis of operation understandable. Many of the details behind the remarkable capabilities of the BIE

---

<sup>†</sup>Email: kmh@lanl.gov, cunning@lanl.gov

can be found in our previous publications [1,3–6], most of which are available on our web site: <http://www.lanl.gov/home/kmh>. Our demonstration will center on several simple 2D tomographic reconstruction examples.

## 2. Historical Background

From the beginning of our development effort, we intended that BIE should be easy to use and should provide a high degree of interactivity with good visualization of the models involved and of the inference process. Additionally, we wanted the application to provide the analyst with a great deal of flexibility in configuring object models and measurement models. After about ten man-years of development, we believe that we have largely met these goals. While the BIE is intended to solve a limited class of problems, that is, radiographic modeling, its underlying design is eminently suited to many other applications.

The basic approach of the BIE is to employ a forward model to calculate the measurements that would be obtained for a given realization of a hypothesized object. Comparison of the calculated and observed measurements is done in terms of the likelihood function, which probabilistically summarizes the differences relative to the statistical uncertainties in the actual measurements. The posterior of the model parameters is obtained by combining the likelihood with a prior probability function that accounts for what is known beforehand.

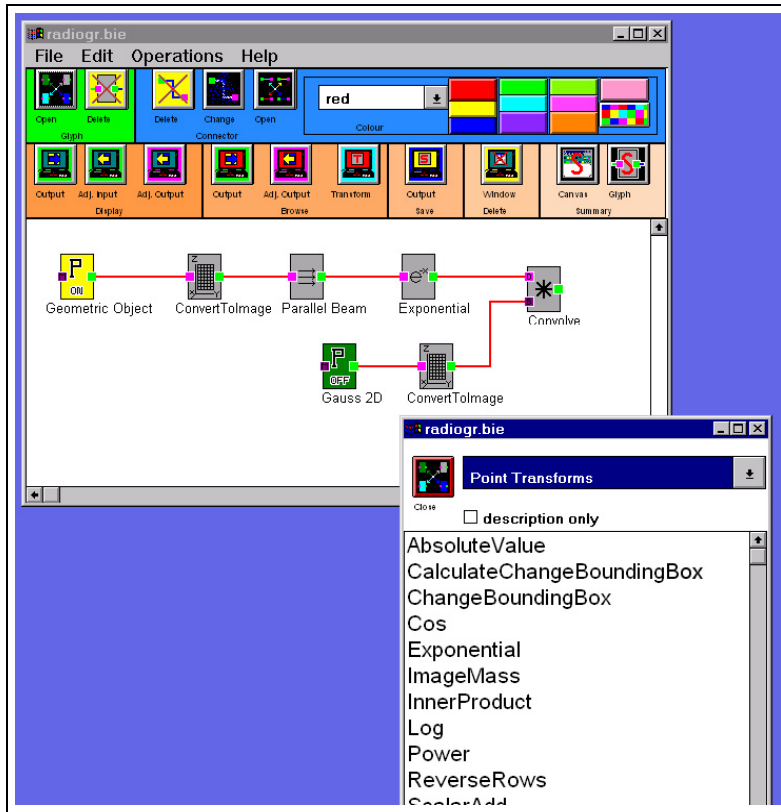
We started developing the BIE in 1993. We embraced from the outset the principles of object-oriented (OO) design, which we felt would result in an application with great flexibility and interactivity. As we have commented before [5], the OO approach to software design has greatly aided and influenced our development. It has had a very positive effect on our approach to algorithm design.

Our assessment was that the OO language SmallTalk80 supplied in the development environment called VisualWorks, supplied by ParcPlace Systems (now ObjectShare<sup>1</sup>) provided the most advanced software-development environment available at that time. Unfortunately, for computationally intensive calculations, SmallTalk is as much as an order of magnitude slower than code written in compiled languages, such as FORTRAN and C. Therefore, SmallTalk is used in the BIE to provide the OO structure essential for the data-flow diagram interface and the data structures. The underlying calculations are coded in FORTRAN and C. Another down side of SmallTalk is that most programmers lack experience with it.

A major thrust in our work has been to learn to cope in a general way with large numbers of parameters, e.g., pixel values, geometrical description, etc. Thus we have avoided numerical methods that are not extendible to numerous variables, which excludes many methods that deal directly with matrices, for example. Our approach is based on dealing directly with numerical valuation of the posterior and its gradient or potentially second-order derivatives. This tack is perhaps less efficient than analytic methods often appropriate for solving small, simple problems,

---

<sup>1</sup>ObjectShare, 16811 Hale Ave., Suite A, Irvine, CA 92606; Web: [www.objectshare.com](http://www.objectshare.com)



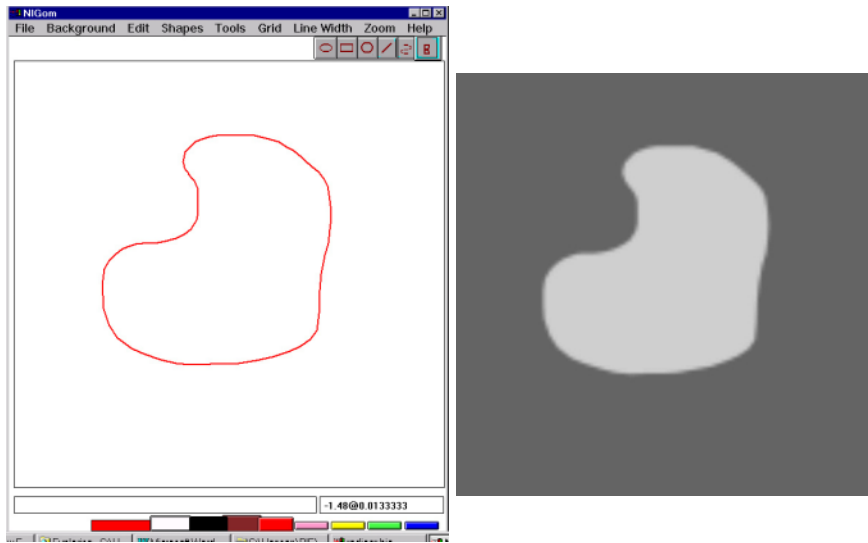
*Figure 1.* The user interacts with the Bayes Inference Engine through its canvas shown here. The set of transforms shown here represent a forward model for a pair of radiographs taken of a physical object that is modeled in terms of its shape. The Glyph Manager window, also shown here at the bottom, provides the selection of transforms that can be placed on the canvas.

but it is able to handle complex models and nonlinear measurement situations, which can not be solved analytically.

### 3. Data Flow Diagram

Models are created in the Bayes Inference Engine through the graphical-programming interface shown in Fig. 1. This means of interaction facilitates understanding of the analysis model. A data-flow diagram is constructed by connecting together transforms, represented visually by rectangles on the canvas [7]. We call these rectangles glyphs, after the nomenclature of the well-known signal-processing application Khoros<sup>2</sup>. While the BIE may appear to be similar to Khoros, its aim and underlying operation differ significantly from Khoros. In fact, we realized early

<sup>2</sup>Khoros Research, Inc., 6200 Uptown Blvd. NE, Albuquerque, NM 87110-4142; URL: [www.khoros.com](http://www.khoros.com)



*Figure 2.* The object created in Fig. 1 as seen a) at the output of the first glyph showing it in terms of its geometrical description, and b) at the output of the second glyph as a gray-scale image, 128 pixels square.

on that the Khoros environment did not meet our needs, which meant that we had to design the BIE from scratch. Some of the important features embodied in the BIE that are not contained in Khoros (at the time we started developing the BIE) include: an object-oriented (OO) programming environment, direct user interaction with the modules represented on the canvas, interaction between the modules that are linked together in the data-flow diagram, and the ability to reverse the flow of data relative to the forward data-flow diagram, which is necessary for adjoint differentiation described below.

When the BIE application is started, a blank canvas is brought up on the computer screen. The analyst creates a data-flow diagram by selecting transforms from a pull-down menu on the Glyph Manager window such as that shown in Fig. 1. The transforms are arranged in categories such as Models and Parameters, Basic Math Transforms, Point Transforms, Priors, Posteriors, and Likelihoods, etc. When a transform is selected, a glyph appears on the canvas. A panel can be brought up to allow the analyst to specify the salient attributes of the transform. Glyphs are linked together by clicking next to the output of one and then next to the input of another, or visa-versa. The connection is visualized as a line drawn between the two glyphs.

The data-flow diagram shown in Fig. 1 represents the calculation of three radiographs of a 2D object. This calculation is based on a forward model of the radiographic process. The leftmost two glyphs calculate a density image of an object in two steps. The first step is to generate a geometrical description of the object boundary. The output of this glyph is shown in Fig. 2a. In this case the

object is represented by a polygon, which is drawn freehand. The second glyph converts the output of the geometrical description to a density image, shown in Fig. 2b.

The remaining glyphs in Fig. 1 correspond to the calculation of three projections of the density image at specified angles, taking the negative exponential to represent x-ray attenuation, and convolution to represent detector blur. In the following analysis, we will neglect the last three steps and work directly with just the unblurred projections. The flexibility of the BIE readily allows extensions to include further details in the radiographic model, including detector sensitivity function, inclusion of known material such as collimators, spectral effects, scattered radiation, beam-source blur, etc. It is possible to save a BIE canvas and subsequently read it back in to replay a computation or resume an analysis.

The output of each glyph can be seen by any one of several modes of interaction. One can double-click on the glyph and the output of the glyph appears. Alternatively, one can see the output by selecting the glyph by clicking on it once and then clicking on the leftmost button in the Display group of hot buttons above the canvas window or by selecting the “display output” option under the Operations button on the window tool bar.

#### 4. Object Modeling and Parameter Estimation

Figure 3 shows the data-flow diagram used to reconstruct an object from the projection data generated by Fig. 1 to which is added uncorrelated Gaussian noise (5% relative to the maximum projection value). The object is modeled in terms of a geometric description of its boundary consisting of a 60-sided polygon, initially in the shape of a circle. We have found that using deformable geometric models [8,9] greatly improves tomographic reconstructions of objects from just a few views [10–12].

In Fig. 3 we use Bayes law in its logarithmic form

$$\varphi = -\log[p(\mathbf{x}|\mathbf{d})] = -\log[p(\mathbf{d}|\mathbf{x})] - \log[p(\mathbf{x})] + \text{constant} \quad , \quad (1)$$

where  $p(\mathbf{x}|\mathbf{d})$  is the posterior for the parameters  $\mathbf{x}$  given the data  $\mathbf{d}$ ,  $p(\mathbf{d}|\mathbf{x})$  is the likelihood, and  $p(\mathbf{x})$  is the prior on the parameters. The likelihood involves calculating the measurements for the given object model, which is handled in Fig. 3 by the same set of glyphs used in Fig. 1 to calculate the projections. The minus-log-likelihood is calculated as  $\frac{1}{2}\chi^2 = \frac{1}{2}\sum_i (y_i - y_{i*})^2/\sigma^2$ , where the  $y_i$  are the actual measurements and  $y_{i*}$  are the corresponding values calculated from the object model, and  $\sigma$  is the estimated rms deviation of the measurement noise. The flexibility of the polygon is controlled by placing a prior on the boundary. The prior used here is an approximation to the integral around the boundary of the square of curvature of the boundary [13], which is calculated in the leftmost glyph along the bottom row of the canvas.

The shape of the object that minimizes the minus-log-posterior is found by the optimizer (the output of the upper-right glyph) by adjusting the vertices of the polynomial boundary of the object. The optimizer operates using the technique described in the next section. The geometry-based reconstruction is shown

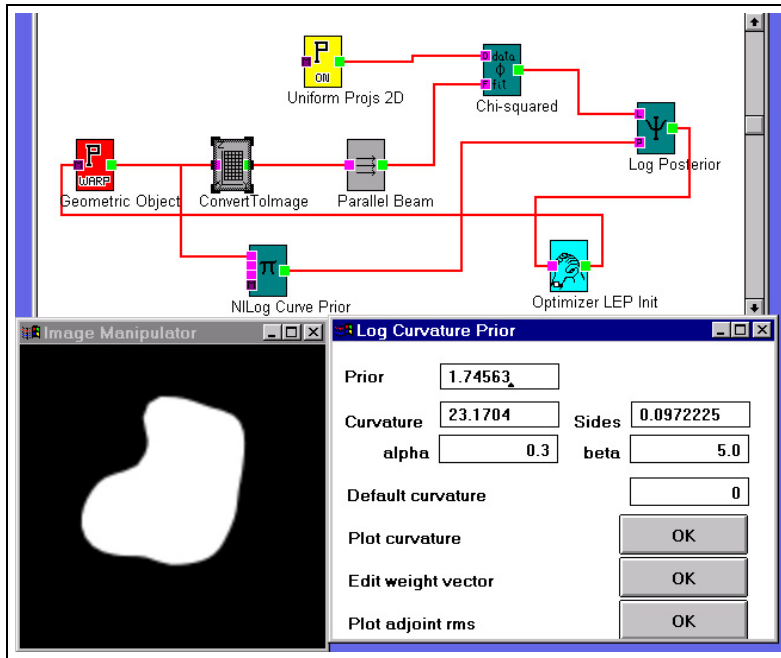
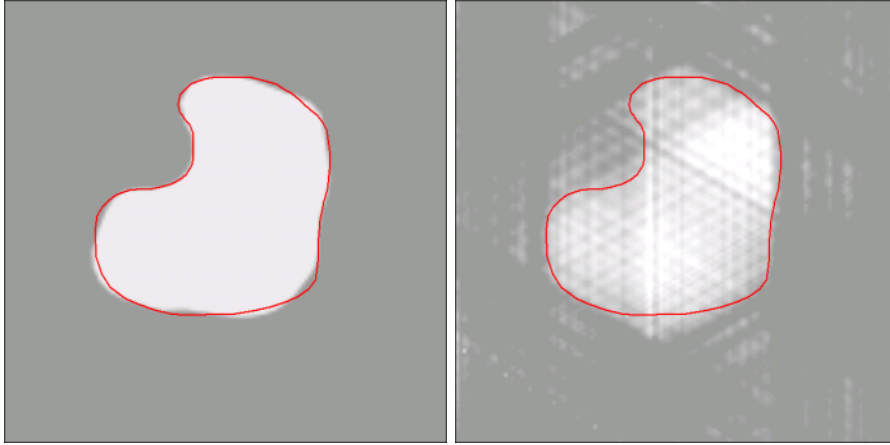


Figure 3. The data-flow diagram to estimate the geometric model parameters, namely the vertices of the polygon, that minimize the minus-log-posterior. The input measurements in the top glyph (Uniform Projs 2D) represent noisy projections of the unknown object from three directions. Also shown on the lower right is the panel specifying the details of the LogCurvaturePrior glyph and, on the lower left, the reconstructed gray-scale image, which is the output of the ConvertToImage glyph.

in Fig. 4a. In the context of the BIE, alternative means of representing the object are easy to try, for example, using Bézier curves instead of polygons, or simply employing a pixelated image, in which case one obtains a standard linear solution to the reconstruction problem. The latter approach results in the reconstruction shown in Fig. 4b. The poor quality of this reconstruction is caused by the extremely limited nature of the data compared to the large number of parameters in the image model, namely  $128^2 = 16384$  pixel values. The generally noisy nature of maximum-likelihood reconstructions is evident, despite imposing a nonnegativity constraint on the image model and the limited number of iterations (20). The noise in this reconstruction only gets worse with more iterations.

#### 4.1. OPTIMIZATION

To obtain the maximum a posteriori (MAP) estimate, we need to minimize the scalar function  $\varphi$ , given in Eq. (1), by varying the parameters that specify 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 employ a technique to calculate these crucial sensitivities, called adjoint



*Figure 4.* Gray-scale renditions of reconstructions from three noisy projections obtained using two different models for the reconstructed object: a) on the left, a geometric model for the object boundary, subject to a prior on the curvature, and b) on the right, a  $128 \times 128$ -pixel image, subject to a nonnegativity constraint on the pixel values. For comparison, the dark curves show the boundary of the original object.

differentiation [14]. 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, especially for optimization.

Adjoint differentiation amounts to implementing the chain rule of differentiation for the computer code. It requires a backward flow of the derivatives of  $\varphi$ , which is easily accomplished in the OO framework [3]. In the BIE, each transform module propagates the adjoint derivative from its output side to its input side. In the code-based approach that we advocate, the adjoint differentiation code is based on the code for the forward calculation. Importantly, the CPU time needed to calculate the derivatives with respect to all the variables in the model is comparable to forward calculation of that model.

We and our collaborators have shown that adjoint differentiation can be beneficially employed in many types of forward modeling codes, for example in a finite-difference code to find the inverse of a time-dependent diffusion problem [15,16] and hydrodynamic codes [17]. See [18] for more details.

With the gradients available, we can perform the optimization using efficient gradient-based algorithms. The algorithm that we use most often is the DFP (Davidon-Fletcher-Powell) quasi-Newton method in which an estimate of the inverse curvature matrix is used to take a Newton step. The inverse curvature matrix is obtained by starting with a diagonal matrix and updating it with each step taken, based on the change in gradient. In the BIE the inverse curvature matrix is not explicitly stored; the accumulated update expression is evaluated each time one needs to multiply a vector by the matrix. This approach maintains our ability to cope with large numbers of variables. The diagonal terms of the starting matrix

can be set by the analyst. To guarantee that the constraints on the parameters are met, we employ the general method of projection onto convex sets (POCS). Each model checks whether its constraints are violated. If they are violated, the parameters are minimally changed to meet the constraints [4].

## 5. Uncertainty Estimation

One of the most important features of Bayesian analysis is that the posterior characterizes the degree of certainty in the models used in an analysis. One of our intentions with the BIE is to provide the means to exploit the full posterior as a measure of uncertainty. The difficulty to overcome is the computational and visualization difficulties of exploring the posterior in a large dimensional space.

### 5.1. MARKOV CHAIN MONTE CARLO

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, as proposed by Skilling et al. [19]. The sequence of images, typically calculated off line, is presented as a video loop. By showing a representative range of plausible solutions, the degree of variability of this presentation provides the viewer with a visual impression of the degree of uncertainty in the inferred model. The present emphasis in Bayesian research on Markov chain Monte Carlo methods [20,21] should yield improved methods to generate random samples of the posterior. Of course, MCMC is more than a tool for visualizing uncertainties; it provides quantitative estimates of the uncertainty in the inferred models. The uncertainty in any aspect of the model may be estimated with respect to any type of uncertainty measure desired, for example, in terms of variance. A notable advantage of MCMC is that the results are obtained with marginalization with respect to any nuisance parameters. It can also be used to estimate the posterior mean (as an alternative to the posterior mode).

To assess the uncertainty for our present problem, the only change required in the optimization canvas, Fig. 3, is to replace the optimizer glyph with an MCMC glyph. The MCMC module in the BIE permits one to save every  $n$ th sample from a sequence generated in a long MCMC run. One can then replay the subsampled MCMC sequence at a later time to scrutinize the model variations. Further details can be found in [13].

### 5.2. PROBING THE MODEL STIFFNESS

A useful alternative to MCMC is based on drawing an analogy between  $\varphi$  and a physical potential [6,22]. Then the gradient of  $\varphi$  is analogous to a force. To explore the reliability of a particular feature of a MAP solution in the BIE, the user specifies it by directly perturbing the selected combination of parameters that characterize that feature. Then, all the parameters are reoptimized to minimize the new  $\varphi$ . The uncertainty in the parameters is indicated by the amount that they move away from their MAP values relative to the strength of the 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 posterior is approximately Gaussian and the parameters are unconstrained [6,22].

## 6. Future Directions

Many extensions to the present capabilities of the BIE are planned. We are developing 3D models and already have very encouraging results [23,24]. A number of our underlying algorithms need improvement. The optimization algorithm is always under development to make it more robust. Recent work [25] has shown the limitations in efficiency of the Metropolis MCMC algorithm. We will explore alternatives that promise improved efficiency. So-called “systematic uncertainties” have always been a bugaboo in the uncertainty analysis of experiments. We think that we can address these in the BIE as highly-correlated uncertainties in the measurements that contribute appropriately to the posterior.

## Acknowledgements

This work has been supported by the United States Department of Energy under contract number W-7405-ENG-36. Many have contributed to the development of the BIE over the years, including Xavier Battle, Roger Bilisoly, George Jennings, Jr., Igor Koyfman, Robert McKee, John Pang, David Wolf, Marion Yapuncich, and Hanji Zhao. We would like to acknowledge the helpful conversations we have had with John Skilling, Steve Gull, Kyle Myers, Robert Wagner, Julian Besag, James Gee, Jim Gubernatis, and Richard Silver.

## References

1. K. M. Hanson and G. S. Cunningham, “A computational approach to Bayesian inference,” in *Computing Science and Statistics* **27**, M. M. Meyer and J. L. Rosenberger, eds., pp. 202–211, Interface Foundation, Fairfax Station, VA 22039-7460, 1996.
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. Faber, D. L. Seth, and A. B. White, Jr., eds., pp. 363–368, Marcel Dekker, New York, 1989.
3. 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.
4. 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.
5. K. M. Hanson and G. S. Cunningham, “The Bayes inference engine,” in *Maximum Entropy and Bayesian Methods*, K. M. Hanson and R. N. Silver, eds., pp. 125–134, Kluwer Academic, Dordrecht, 1996.
6. K. M. Hanson and G. S. Cunningham, “The hard truth,” in *Maximum Entropy and Bayesian Methods*, J. Skilling and S. Sibisi, eds., pp. 157–164, Kluwer Academic, Dordrecht, 1996.

7. 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.
8. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Inter. J. Comp. Vision*, **1**, pp. 321–331, 1988.
9. R. Szeliski, "Probabilistic modeling of surfaces," *Proc. SPIE*, **1570**, pp. 154–165, 1991.
10. 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.
11. K. M. Hanson, "Bayesian reconstruction based on flexible prior models," *J. Opt. Soc. Amer. A*, **10**, pp. 997–1004, 1993.
12. 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.
13. K. M. Hanson, G. S. Cunningham, and R. J. McKee, "Uncertainty assessment for reconstructions based on deformable models," *Int. J. Imaging Systems and Technology*, **8**, pp. 506–512, 1997.
14. 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.
15. S. S. Saquib, K. M. Hanson, and G. S. Cunningham, "Model-based image reconstruction from time-resolved diffusion data," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE*, **3034**, pp. 369–380, 1997.
16. A. H. Hielscher, A. Klose, and K. M. Hanson, "Model-based iterative image reconstruction schemes for time-resolved optical tomography," submitted to *IEEE Trans. Med. Imaging*, 1998.
17. M. L. J. Rightley, R. J. Henninger, and K. M. Hanson, "Adjoint differentiation of hydrodynamic codes," in *CNLS Research Highlights*, Center for Nonlinear Studies, Los Alamos National Laboratory, April, 1998 (WWW: <http://cnls.lanl.gov/Publications/highlights.html>).
18. K. M. Hanson, G. S. Cunningham, and S. S. Saquib, "Inversion based on computational simulations," in *Maximum Entropy and Bayesian Methods*, G. Erickson, ed., Kluwer Academic, Dordrecht, 1998 (to be published).
19. 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.
20. J. Besag, P. Green, D. Higdon, and K. Mengersen, "Bayesian computation and stochastic systems," *Stat. Sci.*, **10**, pp. 3–66, 1995.
21. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London, 1996.
22. 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.
23. G. S. Cunningham, K. M. Hanson, and X. L. Battle, "Three-dimensional reconstructions from low-count SPECT data using deformable models," *Opt. Express*, **2**, pp. 227–236, 1998.
24. X. L. Battle, G. S. Cunningham, and K. M. Hanson, "3D tomographic reconstruction using geometrical models," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE*, **3034**, pp. 346–357, 1997.
25. K. M. Hanson and G. S. Cunningham, "Posterior sampling with improved efficiency," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE*, **3338**, pp. 371–382, 1998.