In Mathematical Perspectives on Neural Networks
P. Smolensky et al., Editors
Lawrence Erlbaum Associates, 1996

Regularization in Neural Nets

Richard Szeliski

Digital Equipment Corporation

Cambridge Research Lab

One Kendall Square, Bldg. 700

Cambridge, MA 02139

(617) 621-6634

szeliski@crl.dec.com *

February 28, 1994

1 Introduction

Regularization is a class of mathematical techniques used in data analysis and engineering to help solve difficult, ill-conditioned estimation and design problems. Regularization was originally applied to problems in numerical analysis such as function approximation and in statistics. The central idea in regularization is to restrict the range of possible solutions to ensure a unique and stable solution. Usually, this is

*This paper is to appear as a chapter in *Mathematical perspectives on neural networks*, Paul Smolensky, Michael Mozer, and David Rumelhart, Eds.

achieved by imposing smoothness constraints on the solution, typically through a penalty term involving derivatives of the solution.

Regularization has been used successfully in computer vision for formulating and solving many low-level vision problems such as stereo matching. It has strong connections to iterative and massively parallel (neural network) algorithms. Recently, regularization has been applied to the formulation and solution of learning problems in neural nets. In this chapter, we review the applications of regularization in computer vision and neural networks, and demonstrate how regularization provides a coherent framework and a useful set of techniques for solving problems which arise in these fields.

The chapter is organized around three main themes: regularization applied to computer vision, the Bayesian interpretation of regularization, and regularization applied to learning in neural networks. We begin the chapter with a review of the classical formulation of regularization applied to the solution of inverse problems, i.e., problems where only the sampled output of a model or process are given, and its unknown parameters must be recovered. We use a one-dimensional curve fitting as our example. We then review how regularization has been applied to computer vision, including the problem formulation, its solution using iterative numerical techniques, and its implementation using neural networks. Next, we discuss how regularization is closely linked to Bayesian estimation techniques, both in finding optimal (and robust) estimates from noisy measurements, and in selecting the optimal amount of smoothing to be used. Turning our attention to learning in neural networks, we show how learning can be interpreted as fitting an approximating function to a limited set of examples, and how regularization can be used both to specify the desired function and to find an efficient representation for it. In addition to these uses of regularization, we also briefly discuss other applications of neural networks to computer vision, and other applications of regularization to neural networks and learning. We close with a discussion of the utility of regularization in both vision and learning, and suggest some promising areas for future research.

2 The regularization of inverse problems

Many difficult problems in the applied sciences and engineering can be formulated or viewed as *inverse problems*, where the task is to recover the input to a complex process, or the unknown parameters governing its behavior, from a limited set of output observations. For example, the recovery of 3-D scene information from 2-D intensity images, one of the central problems in computer vision (Horn, 1977), can be viewed as the inverse of the image formation *forward process*, which has been the subject of extensive study in computer graphics. Similarly, the recovery of 3-D geophysical structure from well logs or from seismic data is a difficult inverse problem (Duda, 1982). In neural networks, the estimation of unknown network weights or parameters from a set of input-output examples (*learning*) is an example of an inverse problem.

Common to all of these inverse problems is the fact that the data usually underconstrains the range of possible solutions. More formally, these problems are said to be *ill-posed* in the sense of Hadamard (Tikhonov and Arsenin, 1977; Bertero, Poggio and Torre, 1988). An ill-posed problem is one whose solution does not exist, is not unique, or does not vary continuously with respect to the input (lack of *stability*).

As a simple example, consider the problem of fitting a function $y = f(x)$ to a set of observations $\{(x_i, y_i), i = 1 \ldots n\}$. The problem is ill-posed since, as stated, it can admit an infinite number of solutions. On the other hand, a solution may not exist if there are noisy, redundant data points (Terzopoulos, 1986b). A variety of techniques could be used to solve this problem, including linear regression (fitting a straight line), fitting low-order polynomials, piecewise linear interpolation, and higher-order spline interpolation (Cybenko, 1994). Many of these techniques work only for certain data or functions, or are not readily extended to higher-dimensional problems.

Regularization is a more general and more widely applicable approach to solving these kinds of problems. In this chapter, we concentrate on methods which minimize a *stabilizing functional* $\mathcal{P}(f)$ in order to find the regularized solution, although many other regularization techniques exist, including methods which restrict the space of possible solutions using low-dimensional approximations to the solution (Tikhonov and Arsenin, 1977). We present below a simplified exposition of regularization theory. For more details, including proofs about existence and stability, see (Tikhonov and Arsenin, 1977; Morozov, 1984), and references therein.

Most commonly, the functional $\mathcal{P}$ measures the *smoothness* of the solution f.[1] For our one-dimensional curve fitting example, a possible stabilizing functional could be

$$\mathcal{P}(f) = \int f_x^2 dx, \tag{1}$$

where f_x denotes the derivative of f with respect to x. Minimizing this smoothness functional while satisfying the original set of equations

$$f(x_i) = y_i, \quad i = 1, \ldots, n \tag{2}$$

results in piecewise linear interpolation. Similarly, if we minimize

$$\mathcal{P}(f) = \int f_{xx}^2 dx, \tag{3}$$

we obtain a piecewise cubic interpolating spline (Ahlberg, Nilson and Walsh, 1967; Schumaker, 1981).

Minimizing $\mathcal{P}(f)$ while satisfying the original set of measurements is just one of three commonly used formulations. Let us denote the original set of measurements by

[1] A functional such as $\mathcal{P}$ maps functions into scalar values.

the system $y = Af$ and write the stabilizing functional as a norm $||Pf||$.[2] In general, $y = Af$ can be used to describe any forward process, e.g., an integral equation of the form $y(x) = \int K(x, \eta) f(\eta)\, d\eta$, where $K(x, \eta)$ can be an arbitrary, spatially varying kernel function. For the set of discrete observations in (2), we have $K(x, \eta) = \sum_i \delta(x - x_i - \eta)$ and $y = \sum_i y_i \delta(x - x_i)$. The linear differential stabilizers in (1) and (3) can we written as $P = \frac{d}{dx}$ and $P = \frac{d^2}{dx^2}$, respectively.

The three main regularization methods (Poggio and Koch, 1985) are:

(i) find the solution f that satisfies $||Af - y|| \leq \epsilon$ and which minimizes

$$||Pf||,$$

(ii) find the solution f that satisfies $||Pf|| \leq C$ and which minimizes

$$||Af - y||,$$

(iii) find the solution f that minimizes

$$||Af - y|| + \lambda ||Pf||.$$

The first of these approaches (with $\epsilon = 0$) corresponds to the interpolating splines introduced above. The second approach, which corresponds to a limited smoothness variation, is not often used and will not be considered further. The third approach, often called the *penalty method*, is the one most commonly used. Here, the *regularization parameter* λ controls the amount of smoothing, with larger λs resulting in smoother solutions. The limit as $\lambda \rightarrow 0$ is the interpolating spline, while the limit as $\lambda \rightarrow \infty$ is a regression with a solution in the null space of P (e.g., linear regression for second order stabilizers). In certain cases, the optimal value of λ can be estimated from the data (see Section 8). This formulation of regularization also has a close correspondence to Bayesian estimation techniques (see Section 7).

[2] A and P, which map functions into functions, are called *operators*.

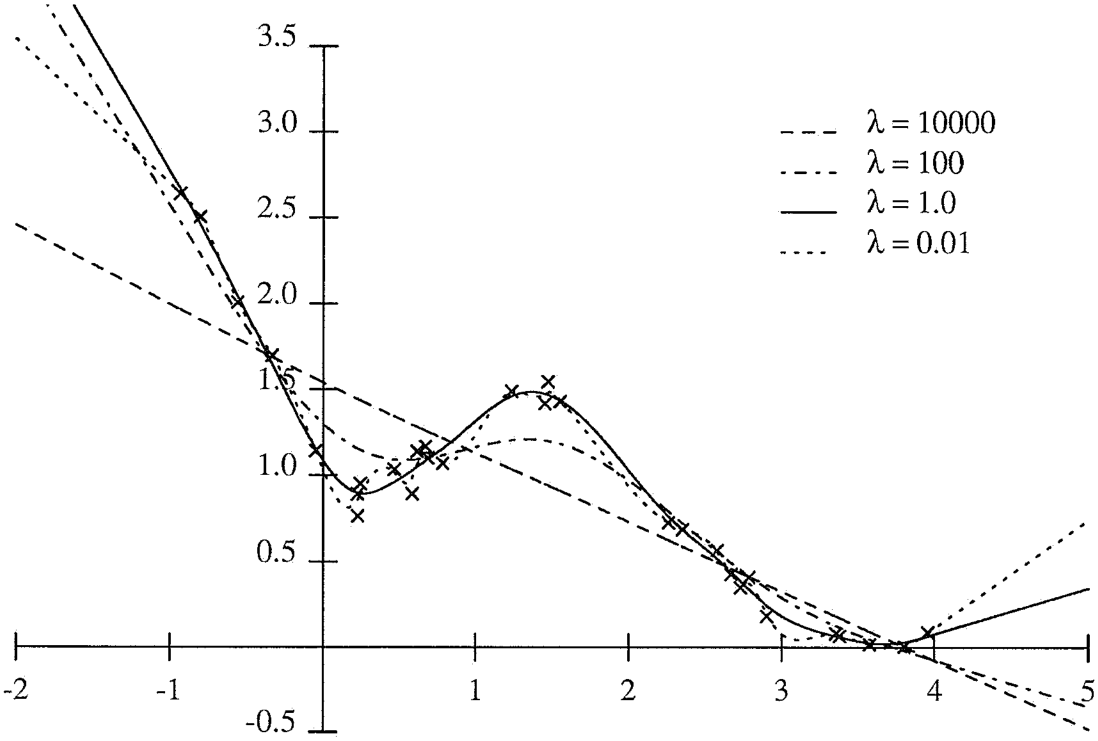

Figure 1: Interpolating/approximating cubic splines

The above curves are obtained by minimizing (4) for various settings of λ. The solution for $\lambda \rightarrow 0$ approaches an interpolating spline (dotted curve). The solution for $\lambda \rightarrow \infty$ is a least-squares line fit (dashed line). Intermediate values of λ control the amount of smoothing. The data points were obtained by randomly sampling a Hermite quadratic function $1.1(1 - x + 2x^2)e^{-x^2/2}$ and adding Gaussian noise with $\sigma = 0.05$ (MacKay, 1992a).

For our data fitting example, the application of the penalty method results in the minimization of the functional

$$\mathcal{E}(f) = \sum_{i=1}^{n} \frac{[f(x_i) - y_i]^2}{\sigma_i^2} + \lambda \int f_{xx}^2 dx, \tag{4}$$

where each squared measurement error $(f(x_i) - y_i)^2$ is inversely weighted by the variance σ_i^2 of the measurement y_i. Minimizing the above functional results in an *approximating* or *smoothing* cubic spline instead of the usual *interpolating* spline (which is the limiting case $\lambda = 0$, see Figure 1) (Reinsch, 1967; Ahlberg, Nilson and Walsh, 1967; Schumaker, 1981).

The second order stabilizer appearing in the second term of (4) is a simple example of the more general pth order stabilizer

$$||f||_p^2 = \sum_{m=0}^{p} \int_{\mathcal{R}} w_m(x) \left(\frac{d^m f(x)}{dx^m} \right)^2 dx, \tag{5}$$

which is also called Tikhonov's stabilizing functional (Tikhonov and Arsenin, 1977). In the above, $w_m(x)$ are spatially varying weighting functions which can be used to locally control the smoothness (and continuity) of the solution (Terzopoulos, 1986b).

Regularization can easily be extended to higher dimensional spaces. One possible generalization of Tikhonov's stabilizing functional to d dimensions is

$$||f||_p^2 = \sum_{m=0}^{p} \int_{\mathcal{R}^d} w_m(\mathbf{x}) |D^m f(\mathbf{x})|^2 d\mathbf{x}, \tag{6}$$

where $D^{2m} = \nabla^{2m}$, $D^{2m+1} = \nabla\nabla^{2m}$, ∇^2 is the Laplacian operator, ∇ is the gradient operator, and $\mathbf{x}$ is the multidimensional domain of the function f (Poggio and Girosi, 1989).[3] This functional is invariant under translation and rotation of the axes

[3] An alternative formula for $|D^m f(\mathbf{x})|^2$ is $\sum_{j_1+\cdots+j_d=m} \frac{m!}{j_1!\cdots j_d!} \left(\frac{d^m f(\mathbf{x})}{dx_1^{j_1} \ldots dx_d^{j_d}} \right)^2$ (Terzopoulos, 1986b).

spanning $\mathbf{x}$, unlike, for example, tensor product splines[4], which are not rotationally invariant. Rotational invariance is an important property when we wish to solve a problem in a coordinate-independent manner.

For spatially invariant weighting functions $w_m(\mathbf{x}) = w_m$, these functionals correspond to sums of *generalized spline functionals* (Duchon, 1976; Meinguet, 1979). Spatially varying $w_m(\mathbf{x})$ can be used to implement *controlled-continuity stabilizers*, which are important for visual surface reconstruction (see Section 3). The regularized solution of problems such as (4) is usually performed by discretization and (potentially parallel) numerical optimization (see Section 4).

One advantage of regularization over other problem formulations is that stabilizers can easily be added to existing problem formulations and, under appropriate conditions, the existence, uniqueness, and stability of solutions can be guaranteed (Tikhonov and Arsenin, 1977; Morozov, 1984). Regularization has been applied to a large number of important problems in low-level computer vision (Poggio, Torre and Koch, 1985) not only to formulate the problems, but also to devise parallel algorithms and analog ("neural") network implementations. Regularization theory has natural and deep connections with Bayesian (statistical) formulations of inverse problems. Such Bayesian formulations can be used to obtain optimal estimates and to characterize the uncertainty in these estimates. Regularization also provides a general framework for learning in neural networks, and for devising efficient learning algorithms. We will explore each of these topics in subsequent sections.

[4] Tensor product splines are functions of the form $f = \sum_{ij} a_{ij} b_i(x) b_j(y)$ that are commonly used in computer aided geometric design (Farin, 1992). They arise in the solution of regularized problems where the stabilizer can be written as the product of univariate stabilizers, e.g., $P = P_x P_y$ (Poggio and Girosi, 1990).

3 Regularization in vision: problem formulation

Computer vision is an interdisciplinary field which studies computational models of visual information processing. Its aims are both to devise efficient algorithms for robotics and machine vision applications, and to investigate computational models of human and animal vision. While these two goals do not always coincide, they provide a fertile ground for interplay and cross-disciplinary research. A number of good textbooks are available, both on computational models of human vision and on robot vision (Marr, 1982; Ballard and Brown, 1982; Horn, 1986; Wechsler, 1990).

In Marr's *hierarchical model of vision* (Marr, 1982), images are processed by independent low-level vision modules to produce a more informative intermediate representation (Figure 2). For this intermediate level representation, Barrow and Tenenbaum (1978; 1981) proposed computing a set of *intrinsic images*, which represent scene characteristics such as distance, orientation, reflectance, and illumination in multiple *retinotopic* (image-aligned) maps. Marr (1978) proposed a *2½-D sketch* which encodes local surface orientation and distance to the viewer as well as discontinuities in the orientation and distance maps.

One of the central problems in the computation of this intermediate representation is *visible surface reconstruction* (Blake and Zisserman, 1987; Terzopoulos, 1988), in which disparate and possibly sparse measurements are integrated into a piecewise continuous representation of depth and orientation (at the input of the 2½-D Sketch box in Figure 2). In its simplest form, this problem reduces to the interpolation of a bivariate function $f(x,y)$ through a set of points $\{(x_i, y_i, z_i)\}$. For example, we may wish to interpolate a depth or elevation map from a set of sparse stereo matches (Grimson, 1981).

Some of the early formulations of this problem were in terms of variational principles (Grimson, 1983). The problem was then reformulated using regularization

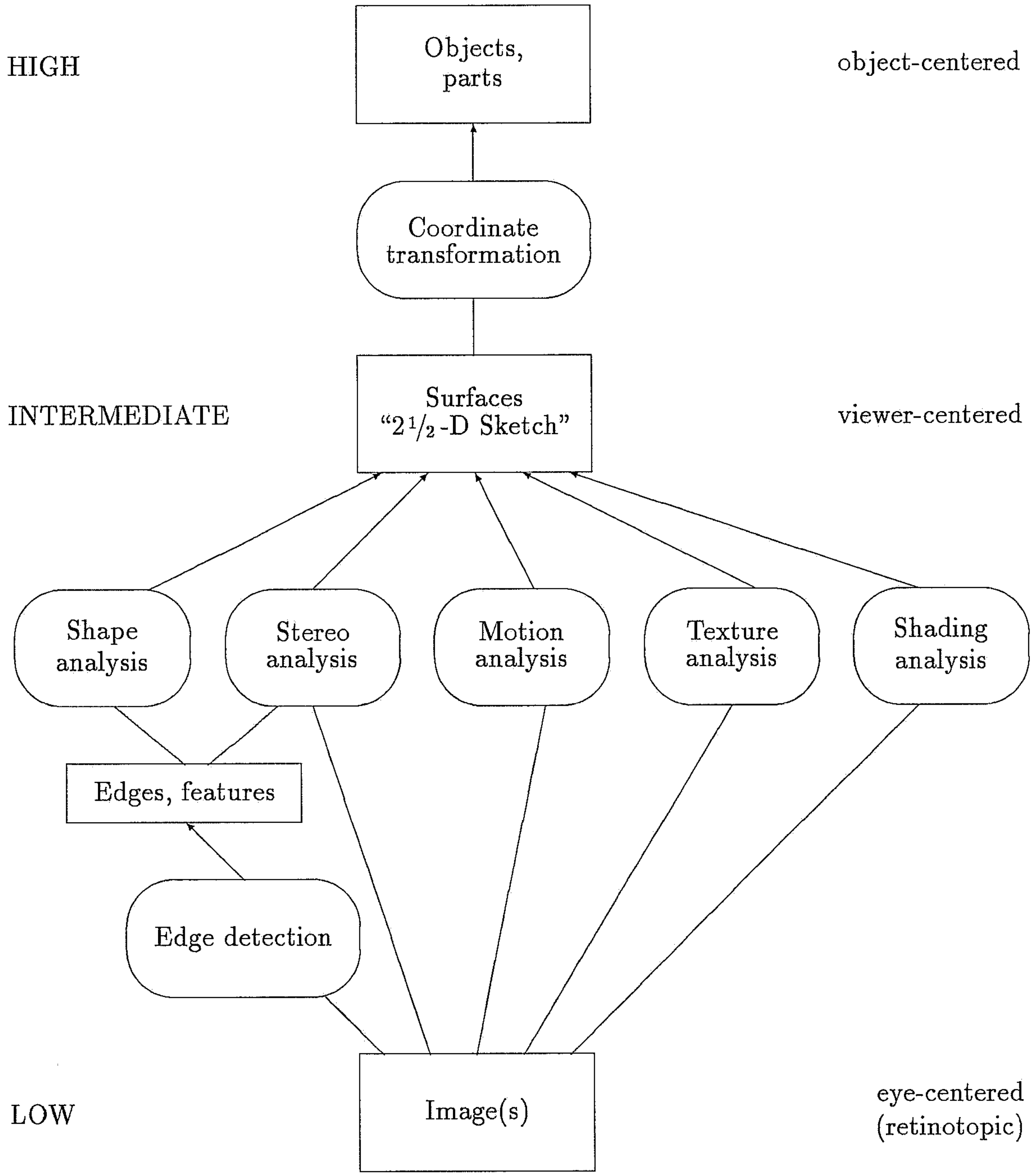

Figure 2: A visual processing hierarchy

The processing of images proceeds from the low level, where features and intrinsic images are extracted, through the intermediate level, where features are grouped and transformed, to the high level, where objects are recognized. Rectangles indicate representations, ovals indicate processes. For alternatives to this bottom-up hierarchy based on independent modules, see (Aloimonos and Shulman, 1989; Clark and Yuille, 1990).

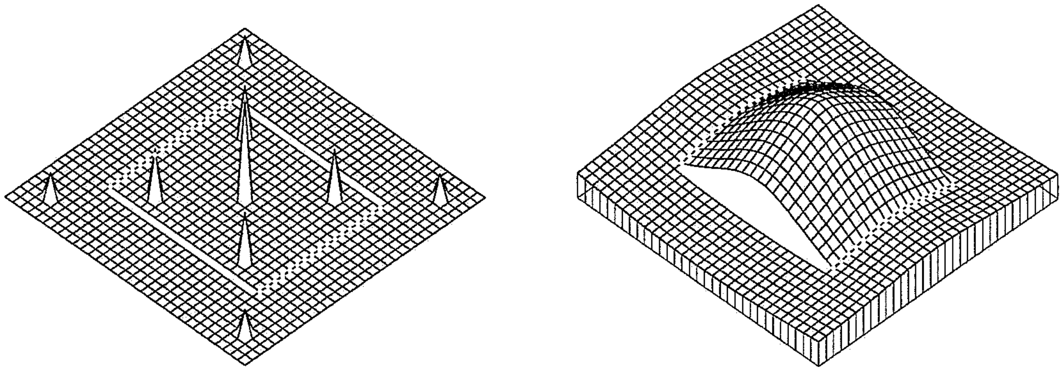

Figure 3: Piecewise continuous surface fitting from 9 data points.
The 9 data points are shown on the left, along with the depth discontinuities (missing line segments) and orientation discontinuities (missing intersections). The discontinuities were introduced by hand for illustrative purposes. The solution on the right clearly shows the effects of depth and orientation discontinuities.

theory (Poggio, Torre and Koch, 1985; Terzopoulos, 1986b). A useful stabilizer for this problem (a special case of (6)) is

$$\mathcal{P}(f) = \int\int \rho(x,y)\{[1-\tau(x,y)][f_x^2 + f_y^2] + \tau(x,y)[f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2]\}\, dx\, dy, \qquad (7)$$

where $\rho(x,y)$ is a *rigidity* function, and $\tau(x,y)$ is a *tension* function (Terzopoulos, 1986b). The rigidity and tension functions can be used to allow depth and orientation discontinuities by setting $\rho(x,y) = 0$ and $\tau(x,y) = 0$, respectively, along some curves in the x-y plane (Figure 3). The solutions which minimize the above smoothness constraint are *generalized piecewise continuous splines under tension* (Terzopoulos, 1986b). While the above example assumes a mixture of first and second order stabilizers, the choice of correct smoothness is in general difficult.

When the locations of the discontinuities (i.e., the zeros of $\rho(x,y)$ and $\tau(x,y)$ functions) are specified, the above functional is quadratic, and the surface reconstruction problem can be solved either directly or with gradient descent techniques (see Section 4). In the more general case, however, the location of the discontinuities must be inferred from the input data itself (Blake and Zisserman, 1987; Terzopoulos, 1988; Mumford and Shah, 1989). This *discontinuity detection* problem is much more difficult than solving the interpolation problem itself.

Fortunately the discontinuity detection problem can also be formulated in a variational (functional minimization) form by making the continuity functions auxillary parameter functions to be estimated along with f using regularization. A variety of formulations for the discontinuity detection problem have been developed, with most of them based on a discretized version of the interpolation problem (Geman and Geman, 1984). For example, a discrete energy E based on (7) using only the first-order stabilizer (i.e., setting $\tau(x,y) = 0$) would be

$$\begin{aligned} E(f,h,v) \;&=\; \sum_{i,j} \Big\{ \frac{1}{\sigma_{i,j}^2}(f_{i,j} - d_{i,j})^2 + \lambda\Big[(f_{i,j} - f_{i-1,j})^2(1 - h_{i,j}) \\ &\qquad + (f_{i,j} - f_{i,j-1})^2(1 - v_{i,j})\Big] + \gamma_{i,j}^h h_{i,j} + \gamma_{i,j}^v v_{i,j} \Big\} \end{aligned} \qquad (8)$$

(Geiger and Girosi, 1991). Here, the $f_{i,j}$ variables are the values of the function $f(x,y)$ sampled on a 2-D rectangular grid, and the $d_{i,j}$ are the input measurements. The $h_{i,j}$ and $v_{i,j}$ variables are the horizontal and vertical *line processes* that indicate the presence ($h_{i,j} = 1$) or absence ($h_{i,j} = 0$) of a depth discontinuity. The line processes lie on a *dual grid* (Figure 4) with respect to the sampled depth values (Geman and Geman, 1984).

In (8), the first term couples the solution f to the input measurements $d_{i,j}$, inversely weighted by the individual measurement uncertainties $\sigma_{i,j}^2$ (where measurements are absent, $\sigma_{i,j}^2 = \infty$). The second term, weighted by λ, corresponds to the discretized version of a first order Tikhonov stabilizer gated by the line processes. The third term penalizes the creation of unnecessary line processes, with $\gamma_{i,j}^h$ and $\gamma_{i,j}^v$ being related to the prior probability of a discontinuity (e.g., as might be provided by running an edge detector over the input image (Gamble and Poggio, 1987; Geiger

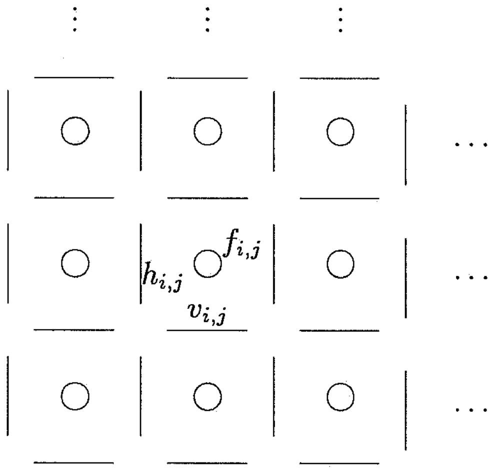

Figure 4: Dual lattice for representing discontinuities

Depth values are represented by circles and discontinuities by line segments. The dual grid on which the discontinuities lie is formed by shifting the original grid vertically and horizontally by half a pixel, resulting in twice as many sites as the original grid.

and Girosi, 1991)). Other discretizations of the surface energy are possible, including ones which have only a single grid of line processes coincident with the depth values (Geiger and Yuille, 1991).

In addition to visible surface reconstruction, regularization has also been applied to a large number of other problems in low-level vision (Poggio, Torre and Koch, 1985; Terzopoulos, 1986a; Bertero, Poggio and Torre, 1988). These include optical flow (motion) computation (Horn and Schunck, 1981; Hildreth, 1986), shape from shading (Horn and Brooks, 1989; Horn, 1990), stereo matching (Witkin, Terzopoulos and Kass, 1987; Barnard, 1989), edge detection (Poggio, Voorhees and Yuille, 1985; Gamble and Poggio, 1987), parametric curve and surface fitting (Kass, Witkin and Terzopoulos, 1988; Terzopoulos, Witkin and Kass, 1988), and motion coherence (Yuille and Grzywacz, 1988). Below, we briefly present the formulations for optical flow, shape from shading, and stereo. We should also note that Markov Random

Field models (see Section 7), which are closely related to regularized formulations, have been proposed as a method for integrating the output of these various low-level vision modules (Poggio, Gamble and Little, 1988; Clark and Yuille, 1990).

Recovering optical flow involves estimating the local pixel motion everywhere in an image given two or more images from a motion sequence. The regularized formulation of optical flow involves smoothing the two dimensional flow field $(u(x,y), v(x,y))$ computed from the temporal and spatial derivatives of the intensity image $I(x,y)$. If the brightness of a portion of an image does not change as it moves, we can write

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = I_x u + I_y v + I_t = 0,$$

(Horn and Schunck, 1981). The regularized solution is obtained by minimizing

$$\int\int (I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2)\, dx\, dy, \tag{9}$$

where the first term is the squared *brightness change constraint* and the second term is the usual Tikhonov stabilizer. The above formulation smooths the flow uniformly in two dimensions. Alternative formulations have been proposed which smooth the flow only along contours (Hildreth, 1986) or perpendicular to the local intensity gradient (Nagel and Enkelmann, 1986).

Shape from shading, or the recovery of surface orientation and depth from the smooth variation in intensity (shading) in an image (Horn and Brooks, 1989), is similarly formulated in terms of the height field $z(x,y)$ and its spatial derivatives $p(x,y) = z_x$ and $q(x,y) = z_y$,

$$\int\int [I - R(p,q)]^2 + \mu[(p - z_x)^2 + (q - z_y)^2] + \lambda(p_x^2 + p_y^2 + q_x^2 + q_y^2)\, dx\, dy, \tag{10}$$

where z, p, and q are estimated separately. In this equation, the first term is the brightness constraint expressed using the non-linear reflectance map $R(p,q)$, the second term is the *integrability constraint* relating the gradient estimates to the height field gradient, and the third term is the usual stabilizer (Horn, 1990). This formulation demonstrates both the application of regularization to a non-linear problem and the use of *mixed methods* where the function $z(x,y)$ and its derivatives $p(x,y)$ and $q(x,y)$ are estimated simultaneously.

As a final example, stereo matching (Barnard and Fischler, 1982) can be formulated as

$$\int\int [L(x+\frac{1}{2}d(x,y),y) - R(x-\frac{1}{2}d(x,y),y)]^2 + \lambda(d_x^2 + d_y^2)\,dx\,dy. \qquad (11)$$

Here, the quantity being estimated is the *disparity map* $d(x,y)$ which measures the relative displacement of features between the left and right input images $L(x,y)$ and $R(x,y)$ (Poggio, Torre and Koch, 1985). This problem is among the most difficult of the regularized low-level vision problems to solve because the cost functional may have many local minima. Both simulated annealing (Szeliski and Hinton, 1985; Barnard, 1989) and continuation methods (Witkin, Terzopoulos and Kass, 1987) have been used to find the global minimum.

4 Numerical solution techniques

In the preceding section, we have seen how regularization can be used to formulate the solution of inverse vision problems as the minimization of a functional. To find the function $f(x,y)$ which attains this minimum on a digital or analog computer, we must first represent it using a finite-dimensional representation (this process is called *discretization*). A convenient way to represent such a function is as a superposition of *basis functions*

$$f(\mathrm{x}) = \sum_i \alpha_i b_i(\mathrm{x}). \qquad (12)$$

For example, a function represented as its Fourier series can be written in this way. The two most widely used classes of basis functions for interpolation and approximation are *kernel splines* and *finite elements.*

Kernel splines are global functions whose superposition exactly minimizes the regularized functional. Assuming that our data constraint is of the form $f(\mathrm{x}_i) = d_i$ or $\min_f \sum_i [f(\mathrm{x}_i) - d_i]^2$, the solution can be written as

$$f(\mathrm{x}) = \sum_{i=1}^{n} \alpha_i K(\mathrm{x} - \mathrm{x}_i) + \sum_{k=1}^{d} \beta_k q_k(\mathrm{x}), \tag{13}$$

which is the superposition of n shifted kernel splines $K(\mathrm{x})$, where n is the number of data points, and d of polynomial functions $q_k(\mathrm{x})$ that span the null space of the stabilizer (Duchon, 1976; Meinguet, 1979; Boult, 1985a). The null space of a stabilizer P is the set of functions f for which $\|Pf\| = 0$. For example, the null space of the second order stabilizer (3), $\mathcal{P}(f) = \int f_{xx}^2 dx$, is the set of linear functions $f(x) = ax+b$, which is spanned by the functions $q_0(x) = 1$ and $q_1(x) = x$.

Deriving equations for the kernel splines is usually only feasible when the regularizer is homogeneous, i.e., $\rho(x, y) = \rho$ and $\tau(x, y) = \tau$ in (7), or $w_m(\mathrm{x}) = w_m$ in (6). For simple regularizers (only one order of derivative) in one dimension, the kernel splines are piecewise polynomial functions (the traditional meaning of *splines*). In two or more dimensions, the kernel splines are more complicated functions of $r = \|\mathrm{x}\|$. For example, the kernel spline $K(r)$ corresponding to the second order stabilizer is $K(r) = r^2 \ln r$ (Duchon, 1976; Meinguet, 1979; Boult, 1985a). Finding the solution that minimizes the functional involves solving a dense system of $(n + d)$ equations in the α_i and β_k. These solutions are related to *kriging* methods in geophysics, which were developed independently based on statistical considerations (Duda, 1982; Hewett, 1986; Hewett, 1988).

Finite element analysis begins by discretizing the domain into a large number of contiguous polygonal regions. The function is approximated over each sub-domain by a low-order polynomial controlled by the values of *nodal variables* which are usually placed at the polygon vertices (Zienkiewicz, 1972). We can also view the finite element representation as the superposition of local piecewise polynomial functions with finite supports (small regions of non-zero value). The resulting discretized energy function is the sum of many small quadratic terms usually involving two or three neighboring nodal variables (Terzopoulos, 1986b; Terzopoulos, 1988). While it is usual to discretize the function so that the nodal variables represent sampled values of the function, we could also represent the function's derivatives with additional nodal variables, which results in *mixed finite element methods* (Harris, 1987; Suter, 1991).

Both kernel splines and finite element models have been used widely in the solution of regularized vision problems. Kernel splines require solving a smaller number of equations, i.e., $(n + d)$ equations, where n is the number of data points and d is the dimensionality of the null space. Kernel splines also give an exact solution, when such a solution exists. Finite element methods can be formulated using a data-independent discretization with m variables, where m is proportional to the number of elements (often, $m \gg n$). On regular grids, finite element methods lead to sparser and more regular sets of equations. They are therefore better suited to massively parallel solutions and neural network implementations (see Section 5). Some further comparisons of these two approaches can be found in (Boult, 1985b; Szeliski, 1989).

Once the discretized energy has been specified, it must be minimized using traditional numerical optimization techniques (Gill, Murray and Wright, 1981; Press et al., 1992). When the discretized energy is quadratic,

$$E(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{A}\mathbf{u} - \mathbf{b}^T\mathbf{u} + E(0), \tag{14}$$

we have the choice of iteratively minimizing this energy, or setting the gradient of the energy to zero and solving the resulting set of equations

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \tag{15}$$

In the above equations, **A** is called the *Hessian* of the energy function (or *stiffness matrix* in finite element analysis), and **b** is called the *residual* (or *force vector*). When the function is not quadratic, the above process must be iterated (using locally quadratic approximations) until a global or local energy minimum is found.

When the number of variables is small or the Hessian matrix of the system is tightly banded, e.g., for energy-minimizing contours or "snakes" (Kass, Witkin and Terzopoulos, 1988), a *direct solution* of the system of equations based on the LU decomposition of **A** may be most efficient (Bathe and Wilson, 1976). For larger systems of equations, e.g., for the two-dimensional surface reconstruction problem, LU decomposition becomes expensive because of *fill in* (George and Liu, 1981), and an iterative minimization algorithm is preferable, even if the energy is quadratic. Simple iterative minimization algorithms include gradient descent and Gauss-Seidel or Jacobi relaxation, where the gradient is divided by the diagonal of **A** to determine the step size (Watrous, 1994). A more efficient form of iterative minimization is conjugate gradient descent, where the direction of descent is modified so that successive directions are *conjugate* with respect to the Hessian **A** (Axelsson and Barker, 1984; Press et al., 1992).

Even with these enhancements, the iterative solution of regularized problems is usually slow. This is because iterative techniques are good at reducing the local (high-frequency) error in the solution, but perform poorly at reducing the global error (Briggs, 1987). A powerful and widely used solution to this deficiency is to represent the field (e.g., surface or image) at a variety of resolutions in a *multiresolution pyramid* (Rosenfeld, 1984). Iterative relaxation can take place at different levels in the pyramid (Figure 5). Efficient techniques for coordinating the relaxation at different levels have been extensively studied in the field of *multigrid analysis* (Brandt, 1977; Brandt, 1981; Hackbusch and Trottenberg, 1982). Multigrid relaxation has been successfully applied to a wide range of regularized vision problems (Terzopoulos, 1986a).

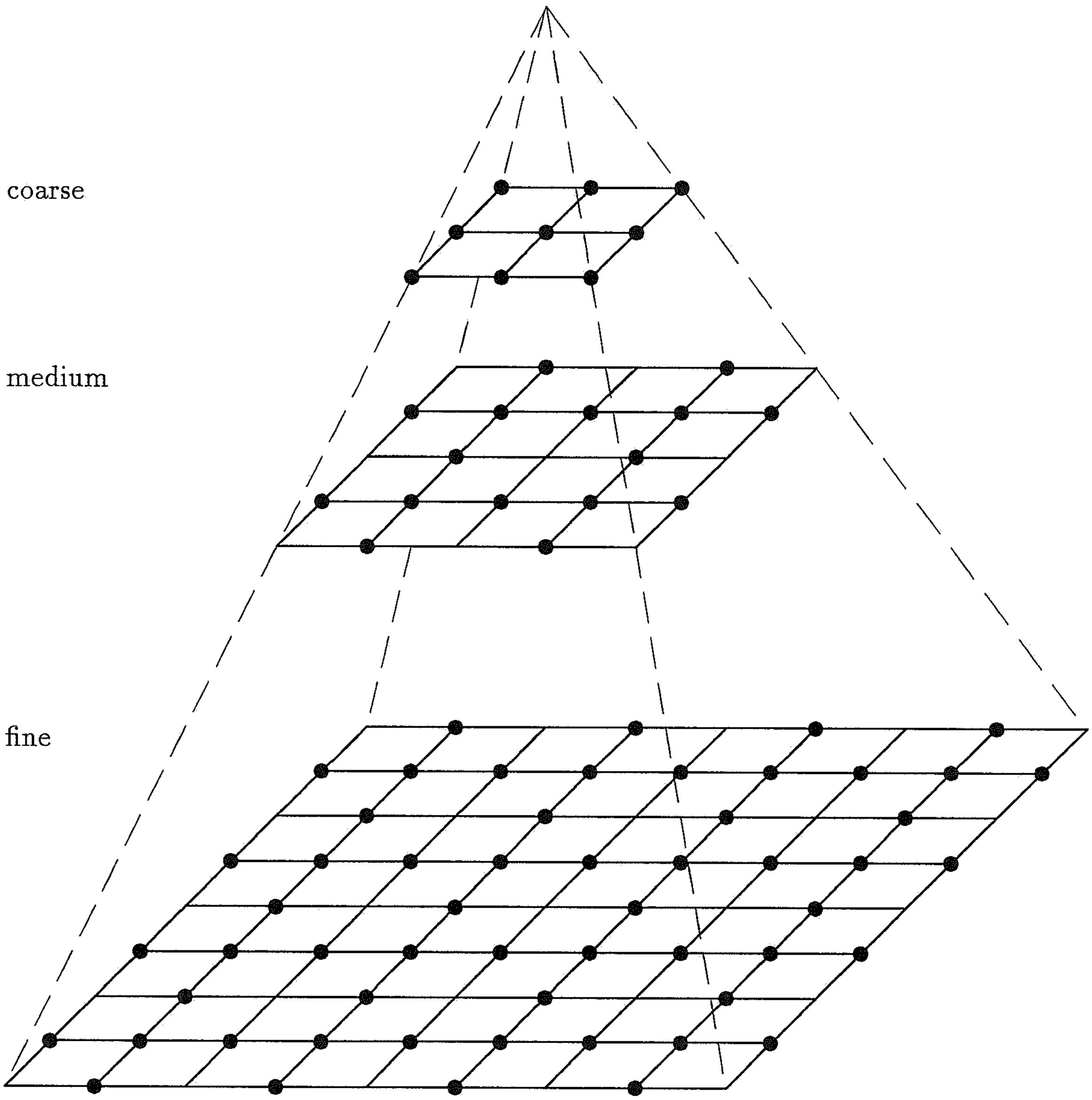

Figure 5: Multiresolution pyramid

The multiple resolution levels are used in the grid correction step and in other more sophisticated multigrid algorithms. When using a hierarchical basis (Yserentant, 1986), the circles indicate the nodes in the multiresolution basis.

An alternative to multigrid, which is somewhat easier to implement, is to use a multiresolution representation to *precondition* the minimization problem (Axelsson and Barker, 1984). By using a change of variables or variable scaling, $\mathbf{u} = \mathbf{S}\mathbf{v}$, the reformulated problem will have better convergence properties. The new system of equations, $\hat{\mathbf{A}}\mathbf{v} = \hat{\mathbf{b}}$, where $\hat{\mathbf{A}} = \mathbf{S}^T\mathbf{A}\mathbf{S}$ and $\hat{\mathbf{b}} = \mathbf{S}^T\mathbf{b}$, has a lower *condition number*, i.e., the ratio of the largest and smallest eigenvalues of $\hat{\mathbf{A}}$, which determines the asymptotic rate of convergence of iterative algorithms, is much smaller that that of the original matrix $\mathbf{A}$.

For regularized problems, a *hierarchical basis*, which is a multiresolution pyramid that preserves the number of original samples, works well (Yserentant, 1986; Bank, Dupont and Yserentant, 1988). Here, the matrix $\mathbf{S}$ maps the distributed multiresolution representation $\mathbf{v}$ back into the original fine-resolution local representation $\mathbf{u}$. This technique has been used in conjunction with conjugate gradient descent to accelerate the solution of visual reconstruction problems (Szeliski, 1990) and non-linear problems such as shape from shading (Szeliski, 1991). For certain regularized problems, e.g., those with sparsely scattered data, the hierarchical basis preconditioner performs even better than multigrid relaxation.

The above discussion applies to problems which have a single local minimum or where the solution can be initialized near the global minimum. In cases where there are many local minima, e.g., in stereo matching, more sophisticated minimization or search techniques must be used.

One technique that fits in well with the regularization approach to inverse problems is *simulated annealing* (Metropolis et al., 1953; Kirkpatrick, Gelatt and Vecchi, 1983; Carnevali, Coletti and Patarnello, 1985; Aarts and Korst, 1994). Instead of always taking a downhill step in energy, simulated annealing algorithms take a random (or *stochastic*) step, with the probability of accepting the step being related to the local energy change through a Gibbs or Boltzmann distribution

$$p(\Delta\mathbf{u}) = \frac{1}{Z} \exp\left(-E(\Delta\mathbf{u})/T\right). \tag{16}$$

This allows the minimization algorithm to escape from shallow local minima. By reducing the amount of randomness (controlled by T) over time, the solution can be forced towards the global minimum (Geman and Geman, 1984). While the usual formulation of simulated annealing involves making probabilistic state transitions based on the local probability density, simulated annealing can also be implemented through *diffusion* (Geman and Hwang, 1986), i.e., by adding a controlled amount of random noise to each gradient descent step. Other related stochastic minimization techniques include *iterated conditional modes (ICM)* (Besag, 1985) and *highest confidence first (HCF)* (Chou and Brown, 1990). The combination of regularization, or related Markov Random Fields (see Section 7), together with stochastic minimization methods has been called *stochastic regularization* (Poggio, Torre and Koch, 1985).

Another popular method for finding global minima is *continuation*, where a non-convex energy function is replaced by a family of parameterized convex functions (Dahlquist and Björk, 1974). For example, in the surface reconstruction problem (8), each binary modulating term $(1 - h_{i,j})$ is replaced by a parameterized convex function $g(h_{i,j})$ and the line variables are allowed to take on real values (Blake and Zisserman, 1987; Terzopoulos, 1988). Recently, Girosi and Geiger (1991; 1991) have shown that such continuation methods can be derived directly from the probabilistic formulation through the use of *mean field theory* (Parisi, 1988). Continuation methods are also widely used in neural networks (Hopfield, 1984; Hopfield and Tank, 1985).

5 Network implementation techniques

The solution of regularized inverse problems, whether in low-level vision, geophysics, or learning, can require massive amounts of computation. Fortunately, the regularized formulation leads naturally to massively parallel algorithms, which can take advantage of either digital parallel architectures or dedicated analog networks. In this section, we explore how such implementations can be developed for low-level vision problems, starting with parallel relaxation algorithms and their implementation on parallel computers, and continuing with analog network solutions and their implementation in analog VLSI.

Neural networks have been used since the beginning of computer vision research for solving both low- and high-level vision problems (Dev, 1974). In computer vision, iterative algorithms based on interconnected computing elements have been called *relaxation* methods, based on earlier work in the numerical solution of partial differential equations. Iterative relaxation has been used both in low-level vision, e.g., using interacting binary units for *cooperative stereo algorithms* (Marr and Poggio, 1976; Szeliski and Hinton, 1985) or multistate units for *relaxation labeling* orientation fields (Hummel and Zucker, 1983), and in higher-level line drawing labeling (Waltz, 1975). The use of numerical relaxation with real-valued units (Horn, 1974; Ullman, 1979; Ikeuchi and Horn, 1981) formed the basis for further developments in variational and regularized formulations of low-level inverse problems (Horn and Schunck, 1981; Grimson, 1983; Hildreth, 1986).

Relaxation algorithms are inherently massively parallel, and their suitability for parallel implementation have long been recognized. Parallel implementations did not become widespread, however, until the advent of commercially available massively parallel architectures (Potter, 1985; Hillis, 1985). Since then, many of the regularized vision problems have been implemented and tested on parallel architectures

(Drumheller and Poggio, 1986; Little, Bulthoff and Poggio, 1987; Little, Blelloch and Cass, 1989). While these implementations have demonstrated the utility of such approaches, they remain too expensive and bulky for widespread use in robotic vision systems.

To overcome these limitations, it has been suggested that special purpose analog networks could be built out of locally connected passive and active electronic elements. Horn (1974) suggested using a grid of resistors to solve the inverse lightness problem. Terzopoulos (1984) designed a simple resistive mesh for solving surface interpolation problems, and Poggio and Koch (1985) extended these ideas to more general regularized vision problems. Parallel analog networks have also been designed for time-varying problems and for more global operations such as moment calculations (Horn, 1988).

The basic idea behind analog nets for low-level vision is shown in Figure 6. This circuit (Harris et al., 1990) implements the same function as the discrete form of the piecewise continuous surface interpolator. The power dissipated by the "vertical" resistor with conductance (inverse resistance) $G = {}^1\!/_2\, \sigma_{i,j}^{-2}$,

$$P = IV = GV^2 = \frac{1}{2\sigma_{i,j}^2}(f_{i,j} - d_{i,j})^2,$$

is the same as the first term of (8). The power dissipated by the "horizontal" resistors whose conductances are $G = \lambda(1 - h_{i,j})$ and $G = \lambda(1 - v_{i,j})$ corresponds to the second term in (8). Since voltages in resistive networks converge towards the state of least power dissipation, the analog circuit effectively minimizes the discrete energy of the surface interpolator.

In the above example, the gating performed by the line processes was folded into the horizontal conductances. In practice, such a circuit could be implemented by putting each resistor in series with a switch (controlled by the line processes $h_{i,j}$ and $v_{i,j}$). However, such a *hybrid* circuit will not solve for the optimal on or off settings

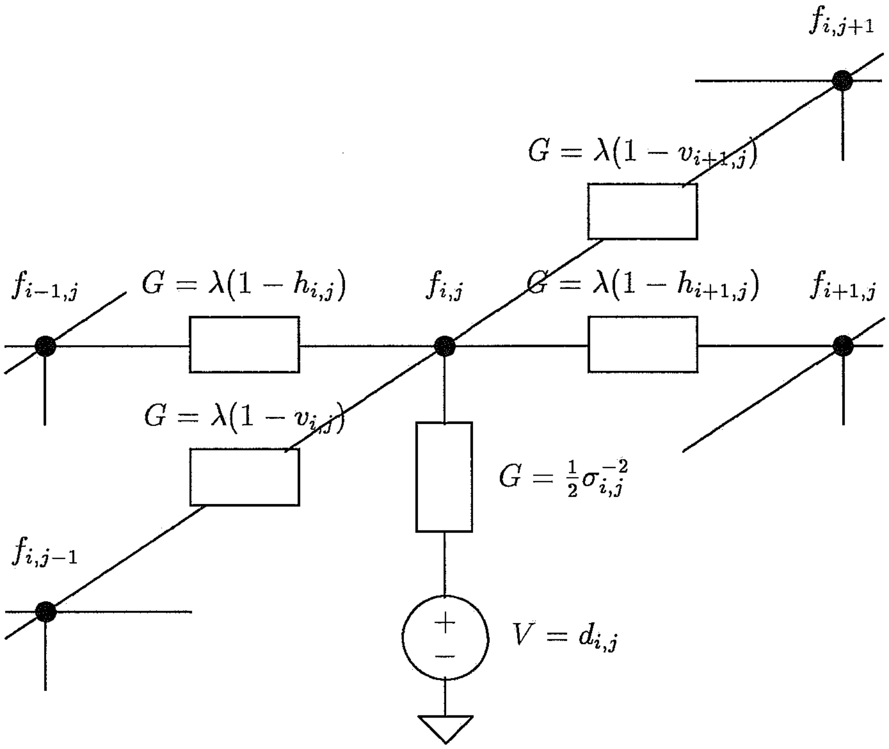
$f_{i,j+1}$
$G = \lambda(1 - v_{i+1,j})$
$f_{i-1,j}$
$G = \lambda(1 - h_{i,j})$
$f_{i,j}$
$G = \lambda(1 - h_{i+1,j})$
$f_{i+1,j}$
$G = \lambda(1 - v_{i,j})$
$G = \frac{1}{2}\sigma_{i,j}^{-2}$
$f_{i,j-1}$
+
−
$V = d_{i,j}$

Figure 6: Analog network for piecewise continuous surface fitting

The input data values $d_{i,j}$ drive the voltage sources $V_{i,j}$ which are attached to the output voltage nodes $f_{i,j}$ through a conductance $G = \frac{1}{2}\sigma_{i,j}^{-2}$. The smoothing is performed by the gated horizontal resistors whose conductances are $G = \lambda(1 - h_{i,j})$ and $G = \lambda(1 - v_{i,j})$. This circuit dissipates the same energy as (8) and hence finds the optimal solution for the regularized interpolation problem.

of the switches. One solution would be to alternate energy (power) minimization in the analog circuit with binary changes in the line process values which decrease the overall power dissipation (Koch, Marroquin and Yuille, 1986; Marroquin, Mitter and Poggio, 1987). Another solution (Koch, Marroquin and Yuille, 1986) is to allow the line processes to take on real values between 0 and 1 and to use Hopfield and Tank's continuation method (Hopfield, 1984; Hopfield and Tank, 1985) to find a solution. The final network contains additional elements which ensure that in the limit, the line processes are driven to 0 or 1.

Simpler circuits for discontinuity detection have been proposed by Harris *et al.* (1990). Their first circuit uses a *resistive fuse* which opens the circuit when the voltage drop across the horizontal resistor exceeds a threshold. Their second circuit is similar, but uses a *finite-gain resistive fuse* whose I-V relationship can be derived from the binary line process formulation using mean-field theory (Geiger and Girosi, 1991). As shown by Geiger and Yuille (1991), this latter circuit is related to other continuation methods such as graduated non-convexity (Blake and Zisserman, 1987) and the continuous network of (Koch, Marroquin and Yuille, 1986). For continuous data, e.g., for *image segmentation*, the networks of Geiger and Yuille can also be related to anisotropic local diffusion processes (Perona and Malik, 1990), which are themselves good candidates for analog network implementation.

The ultimate goal of designing analog networks for computer vision is to implement these networks in high-density analog VLSI (Mead, 1989). The hope is not only to produce useful low-power high-speed processing chips for robotics applications, but also to further study the low-level computation performed by biological systems (Koch, 1989). A number of such circuits have already been implemented and tested. These include a silicon retina which performs center-surround operations (Mead and Mahowald, 1988), a motion (optical flow) detection chip (Hutchinson et al., 1988), and circuits for performing discontinuity detection (Harris, Koch and Luo, 1990;

Harris et al., 1990) and surface interpolation (Suter and Mansor, 1991). Most of these circuits rely on regularization to formulate the problem in a fashion amenable to analog VLSI implementation. A general methodology for analog VLSI design is presented in (Mead, 1989).

Biologically plausible circuits for solving regularized vision problems have also been proposed. These include simple chemical networks (Poggio and Koch, 1985), which could be similar to electronic potentials at the neural level, and more complicated circuits involving pyramidal cells in the visual cortex (Koch, Marroquin and Yuille, 1986). Whether such neural computation is actually taking place in biological vision systems remains to be determined experimentally.

6 Other applications of neural nets to vision

The emphasis in the previous sections has been on low-level vision problems which were formulated using regularization, converted into local, massively parallel iterative algorithms, and implemented in analog VLSI. This represents only a small fraction of the computer vision problems which have been solved with neural networks. In this section, we briefly mention some of these applications. For more examples, the reader is referred to (Wechsler, 1990), which contains a chapter on neural networks in vision.

Many low-level vision problems such as curve detection in textured or cluttered images can be formulated in a local iterative fashion without explicitly resorting to regularization (Zucker, Dobbins and Iverson, 1989). Local parallel algorithms can also be devised that do not require any iteration, e.g., a linear shape from shading algorithm which uses a bank of oriented filters (Pentland, 1989b). Learning has also been applied to low-level vision problems, such as a neural network which learns to classify shaded surfaces by curvature sign (Lehky and Sejnowski, 1988), and to perform texture grouping (Mozer et al., 1992).

Turning to higher-level vision, neural network algorithms have been devised to segment an image into constituent parts through global optimization (Pentland, 1989a; Darrell and Pentland, 1991). Neural network based optimization has also been used to match graphs representing hierarchical object structures (Mjolsness, Gindi and Anandan, 1989). As a final example, radial basis functions (see Section 10) have been used to recognize the identity and orientation of simple wire-frame objects (Poggio and Edelman, 1990).

7 Bayesian interpretation

The solution of inverse problems requires finding a function that is both consistent with the available measurements and "well behaved" in some other sense. Regularization controls the "behavedness" of the function with smoothness measures or other norms on the function. An alternative approach is to view the function recovery process as a *statistical estimation problem*, where probabilistic models of the function and of the measurement process are used to compute statistically optimal estimates.

In such a framework, we model the measurement process as a known linear or non-linear transformation (e.g., subsampling) corrupted by noise. In the general case, we can write this *measurement model*, or *sensor model*, as a conditional probability density $p(d|f)$, where f is the unknown function we wish to recover, and d are the measurements. For example, for two-dimensional surface interpolation, the individual height measurements d_i could be obtained by sampling the function f at locations (x_i, y_i) and adding Gaussian noise whose variance is σ_i^2.[5] This leads to a Gaussian

[5] We allow the variance of each sample to be different for those cases where we have measurements of varying reliability, e.g., the output of a stereo matcher where the variance is inversely proportional to the local contrast. When less is known about the measurement process, we can assume a constant

probability distribution

$$p(d_i|f) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{[d_i - f(x_i, y_i)]^2}{2\sigma_i^2}\right). \tag{17}$$

Assuming that the measurement errors are independent, the conditional probability density $p(d|f)$ can be written as

$$p(d|f) = \frac{1}{Z_d} \exp(-E_d(f, d)), \tag{18}$$

with a *data energy*

$$E_d(f, d) = \sum_{i=1}^{n} \frac{[d_i - f(x_i, y_i)]^2}{2\sigma_i^2} \tag{19}$$

and

$$Z_d = (2\pi)^{n/2} \prod_{i=1}^{n} \sigma_i. \tag{20}$$

The simplest method for recovering the function f, *maximum likelihood estimation*, is to find the function f which maximizes $p(d|f)$. However, this may not produce a unique solution, or it may suffer from other problems associated with ill-posed problems, such as sensitivity to the input. A more powerful approach is *Bayesian estimation*, where a *prior* probability $p(f)$ is hypothesized for the unknown function (Rumelhart, 1994). We can then use *Bayes' Rule* to recover the *posterior* probability

$$p(f|d) = \frac{p(d|f)\,p(f)}{p(d)}, \tag{21}$$

where $p(d)$ is used to normalize the probability density function. Finding the value of f which maximizes the probability density produces the *maximum a posteriori* (MAP) estimate.

To develop this model further, let us examine the case of surface reconstruction. We describe the discrete model here, since it is simpler to explain. For a description of the continuous model, the *pseudo-Markovian field*, see (Adler, 1981). To obtain

variance, $\sigma_i^2 = \sigma^2$.

good surface reconstructions, we need a prior model which favors piecewise smooth surfaces. A good model for such a surface is a *Markov random field* (MRF), where the conditional probability of a value $f_{i,j}$ depends only on the state of its immediate neighbors (Geman and Geman, 1984). Such a distribution can be written as a Gibbs or Boltzmann distribution

$$p(f) = \frac{1}{Z_p} \exp(-E_p(f)/T_p), \tag{22}$$

where $E_p(f)$ is a *prior energy* function made up by summing local *clique energies*

$$E_p(f) = \sum_{c \in C} E_c(f),$$

C is the set of all cliques (interacting variables), and

$$Z_p = \sum_f \exp(-E_p(f)/T_p)$$

is called the *partition function* (Geman and Geman, 1984). The *temperature* parameter T_p controls the "peakedness" of the distribution and, as we will see shortly, is related to the regularization parameter λ. In our surface interpolation example, the MRF energy for the prior probability is

$$E_p(f) = \sum_{i,j} \left[(f_{i,j} - f_{i-1,j})^2 (1 - h_{i,j}) + (f_{i,j} - f_{i,j-1})^2 (1 - v_{i,j}) \right], \tag{23}$$

which represents the summed local smoothness energies of a piecewise continuous surface.

Combining the sensor model (18) and (19) with the prior model (22) and (23) using Bayes' Rule (21), we obtain the posterior probability density function

$$p(f|d) = \frac{1}{Z} \exp(-E(f)), \tag{24}$$

with

$$E(f) = E_d(f, d) + \frac{1}{T_p} E_p(f). \tag{25}$$

This *posterior energy* corresponds exactly to the discrete energy of the regularized surface interpolation problem (8), if we set $\lambda = 1/T_p$ and assume appropriate prior probabilities for the $h_{i,j}$ and $v_{i,j}$ line processes (Geiger and Girosi, 1991).[6]

We therefore have the result that maximizing the *a posteriori* probability is equivalent to minimizing the regularized energy, under the assumption of Gaussian noise and certain specific priors (Kimeldorf and Wahba, 1970; Poggio, Torre and Koch, 1985; Marroquin, Mitter and Poggio, 1987; Szeliski, 1987; Szeliski, 1989). Since the two approaches will produce the same solution, is there any advantage to a probabilistic formulation? Indeed, there are several, which we summarize below and then examine in more detail.

First, the statistical assumptions corresponding to various smoothness constraints and data constraints can be explicitly stated and examined (Szeliski, 1989). Second, alternative estimates based on different cost functions (statistical estimators) can be computed by finding the function f^* which minimizes $\int C(f, f^*)p(f|d)df$, where $C(f, f^*)$ is a cost function (Marroquin, Mitter and Poggio, 1987).[7] Third, the uncertainty in the posterior model can be quantified by computing confidence intervals for the regularized solution. Fourth, Bayesian modeling can be used to optimally integrate multiple measurements over time (Matthies, Kanade and Szeliski, 1989). Lastly, optimal values for the unknown regularization parameters, such as λ, can be computed using statistical considerations.[8]

[6] When we have a uniform data variance $\sigma_i^2 = \sigma^2$, the values of σ and T_p in (25) are "redundant" in the sense that scaling both by the same factor results in the same MAP estimate. However, changing these values affects the variance of the final estimate, which may be an important quantity to compute (see below).

[7] The same result cannot in general be obtained by simply modifying the energy functions (Marroquin, Mitter and Poggio, 1987).

[8] The Bayesian approach does not have any performance disadvantages compared to optimization (regularization) approaches, since it subsumes the latter techniques. The approach is, however,

The ability to explicitly state the statistical assumptions underlying regularization is one of the chief attractions of the Bayesian approach to inverse problems. It has long been known that regularization is equivalent to assuming a particular covariance structure for the field being estimated (Kimeldorf and Wahba, 1970; O'Sullivan, 1986; Terzopoulos, 1986b). By analyzing the power spectrum of such a correlated field, it can be shown that for simple regularizers (with only one non-zero w_m in (5) or (6)), the spectrum is fractal, i.e., self-similar over scale (Szeliski, 1987).[9] The ability to devise better data constraints by systematically modeling the forward transformations and sources of noise in sensors is of potentially even greater impact (Szeliski, 1989; Clark and Yuille, 1990). In its simplest form, this shows up as the weighting of measurements by their inverse variance in (4) and (8). In more sophisticated examples, sensor models can be constructed that solve the correspondence problem between the data and unknown surface, and reject outlier data points (see Section 9).

Since a Bayesian model in theory specifies a complete posterior distribution, we are not limited to choosing the most likely (MAP) estimate, which corresponds to the minimum energy state in a regularized problem. Indeed, for certain low-level vision problems such as image reconstruction, there is evidence that finding the minimum variance estimator, which corresponds to the *maximizer of posterior marginals* or MPM, performs better than MAP estimation because it minimizes the average number of misclassified pixels or the average error, rather than looking for the single most

sometimes more cumbersome to formulate and implement. Some statisticians also cite the need to devise prior probabilities as a weakness of the Bayesian approach (Wolpert, 1993).

[9] The energy in (6) for constant $w_m(\mathrm{x}) = w_m$ can be rewritten in the frequency domain using Pareseval's Theorem as $E(F(\boldsymbol{\omega})) = \int |H_p(\boldsymbol{\omega})|^2 |F(\boldsymbol{\omega})|^2 d\boldsymbol{\omega}$ where $|H_p(\boldsymbol{\omega})|^2 = \sum_m w_m |\boldsymbol{\omega}|^{2m}$ and $\boldsymbol{\omega}$ is the multidimensional frequency vector (Szeliski, 1987). The prior distribution for $f(\mathrm{x})$ is therefore correlated Gaussian noise with a spectrum $S_f(\boldsymbol{\omega}) = |H_p(\boldsymbol{\omega})|^{-2}$. If only one of the w_m is non-zero, this spectrum is fractal, i.e., the signal is self-similar (in a stochastic sense) over all scales (Mandelbrot, 1982).

plausible solution (Poggio, Torre and Koch, 1985; Marroquin, Mitter and Poggio, 1987).[10] Alternatively, the cost function whose expected value is being minimized can be used to bias the solution towards high-level goals, e.g., by preferring distance underestimates in a collision detection application. The statistical interpretation also enables the development of alternative optimization algorithms such as *highest confidence first*[11] (HCF) (Chou and Brown, 1990) and mean-field approximations (Geiger and Girosi, 1991).

Bayesian models based on regularization can be used to assign a confidence or certainty to the posterior estimate. The simplest way to do this is to draw *error bars* or *confidence intervals* around the estimated solution (Wahba, 1983; Szeliski, 1989; Keren and Werman, 1990; MacKay, 1992a). Figure 7 shows the error bars corresponding to the interpolating curve with $\lambda = 1.0$ in Figure 1. Notice how the error bars increase in areas where the data is less dense, indicating a decreased certainty in the estimated solution.

In theory, the second order statistics (variance and covariance) of the posterior estimate could also be computed, but for large systems, this may be computationally infeasible.[12] Consider for example the discrete version of a quadratic energy functional (14), $E(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{A}\mathbf{u} - \mathbf{b}^T\mathbf{u} + E(0)$, whose MAP and minimum variance estimates are both $\mathbf{u}^* = \mathbf{A}^{-1}\mathbf{b}$, and whose covariance matrix is $\mathbf{A}^{-1}$. While $\mathbf{A}$ is usually sparse

[10] The minimum variance estimator, also known as the *minimum mean squared error* (MMSE) estimator, minimizes the expected value of the cost function $C(f, f^*) = |f - f^*|^2$, i.e., the expected squared error. The maximizer of posterior marginals independently estimates the mean value for each component of f (Marroquin, Mitter and Poggio, 1987).

[11] The highest confidence first algorithm selects the variable which has the highest confidence in its estimate and freezes its value. It then proceeds in a similar fashion to estimate all of the other variables (Chou and Brown, 1990).

[12] However, we can draw random samples from the posterior distribution to illustrate the range of possible solutions (Szeliski, 1987; MacKay, 1992a).

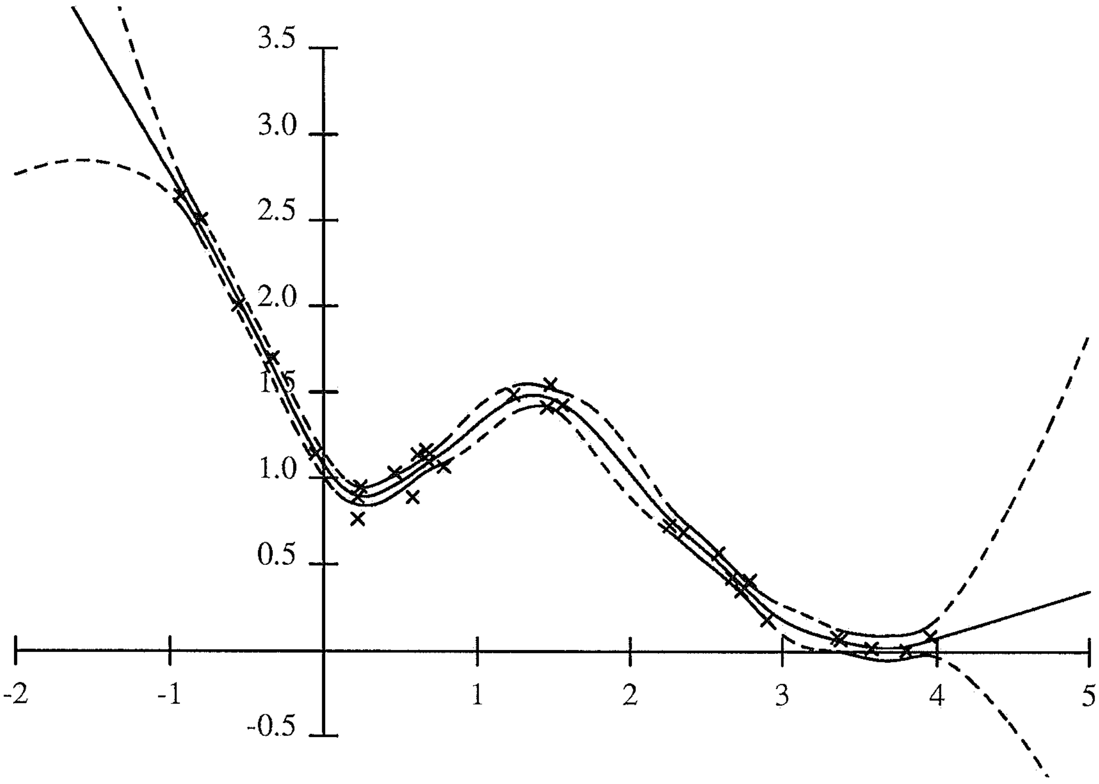

Figure 7: Cubic spline with confidence interval

The confidence envelope was obtained by computing the posterior variance at each location x_i as the solution to the interpolation problem $\{(x_j, \delta_{ij}\sigma_i^2), j = 1 \ldots n\}$. These variances were converted into standard deviations $\sigma(x)$ by taking a square root. The dashed envelopes represent the curves $f(x) \pm 2\sigma(x)$. Values at intermediate points between original data samples were computed by introducing extra measurements with infinite variance (Szeliski, 1989). An alternative technique, based on a basis function expansion, is presented in (MacKay, 1992a). Note how the variance (width of the envelope) increases away from the known data points.

and banded, and can thus be stored even for large systems, the covariance matrix $\mathbf{A}^{-1}$ is not, and cannot therefore be explicitly computed for systems with thousands of variables (such as those which arise in low-level vision).

The existence of a posterior distribution allows the current estimate to be integrated with new measurements in an *incremental* or *recursive* fashion. Such recursive estimation algorithms are often based on the *Kalman filter* formalism (Gelb, 1974). In such a formulation, the posterior estimate in one iteration $p(f^{(k)}|d^{(k)})$ becomes the prior distribution for the next iteration $p(f^{(k+1)})$ (where k is the iteration number). However, careful attention must be paid to the exact choice of formulation in order to produce computationally feasible algorithms (Szeliski, 1989). This approach has been used to combine a regularized formulation of optic flow with a sequence of images taken from a translating camera to obtain depth estimates whose quality improves over time (Matthies, Kanade and Szeliski, 1989).

8 Parameter estimation

The final application of the Bayesian formulation, namely parameter estimation, is of sufficient importance that we will devote a separate section to it. Indeed, the ability to estimate the unknown model parameters such as λ from the data itself is often cited as one of the chief attractions of regularization.

Many of the proposed techniques for estimating λ do not explicitly rely on a Bayesian formulation, although most do have some basis in statistics. For example, Anderssen and Bloomfield (1974b; 1974a) use spectral techniques to determine the shape of a smoothing filter. Wahba and her colleagues have developed a number of techniques called *cross-validation* and *generalized cross-validation* (Craven and Wahba, 1979; Wahba and Wendelberger, 1980; O'Sullivan, 1986). The basic idea is to fit the function with some of the data points removed, and to then measure the discrepancy between the fitted curve and the reserved data points. Both oversmoothing and undersmoothing (which leads to wildly oscillating results) will generate a large discrepancy. The regularization parameter λ is set to the value which minimizes the generalized cross-validation (GCV) measure

$$V(\lambda) = \sum_{i=1}^{n} w_i(\lambda)[f_\lambda^{(i)}(x_i) - y_i]^2$$

where $f_\lambda^{(i)}(x)$ is the curve fit to all of the data points except for i, and $w_i(\lambda)$ is a weighting function defined in (Craven and Wahba, 1979).

Bayesian approaches to the estimation of λ are based on maximizing the probability of observing the data

$$p(d) = \int p(d|f)p(f;\lambda)df$$

where the shape (tightness) of the $p(f;\lambda)$ distribution is a function of λ ((22) with $\lambda = 1/T_p$). While this may at first seem hopeless since f is a family of functions, the discrete version of the problem with quadratic functionals corresponds to multivariate Gaussian distributions. The resulting minimization can be expressed in terms of quadratic energy terms and logarithms of matrix determinants (Szeliski, 1989; MacKay, 1992a; Wolpert, 1993).

9 Robust statistics and the assignment problem

The preceding discussion of Bayesian models has assumed that the prior and sensor probability distributions are known and accurate. In practice, this is seldom the case. To overcome this deficiency and to ensure that our algorithms are less sensitive to gross errors and outliers, we should use ideas from *robust statistics* (Huber, 1981; Hampel et al., 1986). In its simplest form, robust statistics suggests that quadratic penalty terms such as those in (2), (8), or (19) be replaced with functions that penalize extreme values (*outliers*) less severely. We have already seen this concept applied to discontinuity detection in Section 3, where the quadratic penalty for the inter-neighbor difference is replaced by a non-quadratic function (Blake and Zisserman, 1987; Geiger and Girosi, 1991). Similar ideas can be applied to the data penalty terms, which effectively replaces the "springs" tying the data to the surface in (4) and (19) with "breakable" or "weak" springs of the form

$$E_d(f,d) = \sum_{i=1}^{n} g(d_i - f(x_i, y_i)), \tag{26}$$

where $g(x)$ is a function which penalizes large differences less than a quadratic function (Szeliski, 1989; Harris, Liu and Mathur, 1991). Robust techniques can also be applied to the complete surface fitting process itself (Sinha and Schunck, 1990) and to neural networks learning (Hinton and Nowlan, 1990; Nowlan and Hinton, 1992; Martin and Connor, 1993). For a more detailed description of alternative robust statistical techniques, the reader is referred to Huber (1981) and Hampel *et al.* (1986). While traditional robust statistics assume a particular form of long-tailed distribution, an even more sophisticated approach is possible where the shape of the distribution is itself adjusted based on the data (Box and Tiao, 1968; Box and Tiao, 1973).

In certain problems, replacing the data constraints with robust error measures is itself not sufficiently general, since the *assignment*, or *correspondence*, between the data and surface may itself be unknown. Consider for example the fitting of an elastic 3-D parametric surface (Terzopoulos, Witkin and Kass, 1988) to a collection of 3-D data points.[13] Or as a simpler example, consider fitting a deformable 2-D contour (Kass, Witkin and Terzopoulos, 1988) to a set of points in the plane. This latter example has a strong connection to an analog formulation of the Travelling Salesman Problem (Durbin and Willshaw, 1987) and to the general problem of 2-D curve

[13] A version of this problem with known correspondence is the fitting of a digital elevation map $f(x, y)$ to a set of height measurements $d_i = f(x_i, y_i)$.

matching (Hinton, Williams and Revow, 1992). In such an approach, the pairwise constraint between each data point and an identified point on the curve is replaced by a general *force field* which attracts nearby curve points toward each data point. It can be shown that this force is equivalent to assuming that the data points were generated by a noisy sampling of an unknown point along the curve (Durbin, Szeliski and Yuille, 1989), thus giving a Bayesian justification for the heuristic optimization algorithm. Similar ideas can be used to develop more general sensor models (Szeliski, 1989) and to convert discrete matching problems into continuous optimization formulations (Yuille, 1990). These techniques extend the range of applicability of regularization to problems which may not at first appear to have a suitable structure.

10 Learning as function approximation

In the final part of this chapter, we address the task of learning input-output mapping from a limited set of examples, which is one of the central problems in current neural network research.

As has been shown recently by Poggio and Girosi (1990), there exists a deep connection between certain classes of neural networks and the solution of approximation problems formulated using regularization (see also Cybenko (1994)). In particular, we will show how radial basis functions arise naturally in the solution of regularized problems, and how these functions can be implemented using simple neural networks (Broomhead and Lowe, 1988; Moody and Darken, 1989). These results allows us to specify the goal of a feedforward neural net, i.e., finding a smooth input-output mapping with good generalization properties, in a manner that is independent of the underlying neural network architecture.

Given a collection of input/output pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, where $\mathbf{x}_i$ and possibly $\mathbf{y}_i$ are vector valued, the *learning* problem is formulated as constructing a (usually non-linear) network which will reproduce $\mathbf{y}_i$ given $\mathbf{x}_i$, and will also *generalize* well to other values of $\mathbf{x}_i$ (as in approximation theory, the meaning of "generalize well" can be ill-defined). A natural way to view this task is to search for a function $\mathbf{f}(\mathbf{x})$ such that $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$ which varies smoothly in the vicinity of the $\mathbf{x}_i$ samples (alternative formulations may involve finding a function $f(\mathbf{x}, \mathbf{y})$ such that $f(\mathbf{x}_i, \mathbf{y}_i) = 0$, but this requires the system to be given negative examples as well). The problem of learning can thus be formulated as one of finding a good approximating function.

The problem of extrapolating a complete multidimensional function from a number of discrete samples has been extensively studied in approximation theory (Laurent, 1972). A simple example of this is the two-dimensional surface reconstruction problem which we studied in Section 3. A popular class of approximating functions is based on the superposition of shifted *radial bases* $G(\|\mathbf{x}\|)$ and a small number of polynomial functions $q_k(\mathbf{x})$,

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i G(\|\mathbf{x} - \boldsymbol{\xi}_i\|) + \sum_{k=1}^{d} \beta_k q_k(\mathbf{x}) \tag{27}$$

(Powell, 1987) (note that f, α_i, and β_k can be vector valued).[14] When the number of data points is not too large, it is customary to center one basis function on each data point $\boldsymbol{\xi}_i = \mathbf{x}_i$. Conditions under which an interpolating surface exists for such a collection of radial bases and data points have been derived (Micchelli, 1986).

The choice of basis functions can be made based on previous experience or computational simplicity, or it can be derived directly from a regularized formulation of the problem. For example, the *generalized splines*, which we studied in Section 2, have a kernel spline representation which uses radial bases (Duchon, 1976; Meinguet, 1979) (see (13) in Section 4). In general, the shape of the basis function $G(\|\mathbf{x}\|)$ and of the polynomial functions $q_k(\mathbf{x})$ can be derived from the Green's function and the

[14] The q_k serve to model the global characteristics of the solution, e.g. a pervasive constant, linear or quadratic bias.

null space, respectively, of the regularization operator P (Poggio and Girosi, 1989).[15] A particularly interesting example is the choice of $w_m = \sigma^{2m}/(m!\,2^m), m = 0, \ldots, \infty$, in (5) which results in Gaussian radial bases with a standard deviation of σ

$$G(\|\mathbf{x}\|) = \exp \frac{-\|\mathbf{x}\|^2}{2\sigma^2}$$

(Yuille and Grzywacz, 1988).[16]

Radial basis functions can be directly implemented by a three-layer neural network, as shown in Figure 8 (Broomhead and Lowe, 1988; Moody and Darken, 1989). When the basis centers $\boldsymbol{\xi}_i$ are fixed, the solution for the output weights α_i can be computed using least squares fitting (Broomhead and Lowe, 1988). When the centers are variable and to be determined during learning, gradient descent techniques or other non-linear optimization techniques become necessary (Moody and Darken, 1989). This latter approach can take advantage of local parallelism and is closer in structure to other iterative learning algorithms such as *backpropagation* (Rumelhart, Hinton and Williams, 1986). In addition to the output weights and the basis centers, the widths of the basis functions can also be adjusted (Moody and Darken, 1989), and the number of basis units can be modified during learning (Platt, 1991). The most general version of radial basis functions are *hyper basis functions* (HBF), or *regularization networks*, which use a combination of different sized kernels, movable

[15] When the w_m are constant functions, the Fourier transform of the basis function $G(\mathbf{x})$ is

$$\mathcal{F}\{G\} = \left[\sum_{m=0}^{\infty} w_m |\boldsymbol{\omega}|^{2m}\right]^{-1}$$

(this is identical to the power spectrum of the prior model), and the null space is the solution to the partial differential equation

$$\sum_{m=0}^{\infty} w_m |D^m f(\mathbf{x})|^2 = 0.$$

[16] $\left[\sum_{m=0}^{\infty} \frac{\sigma^{2m}|\boldsymbol{\omega}|^{2m}}{m!2^m}\right]^{-1}$ is the Taylor series expansion of $\left[\exp \frac{\sigma^2|\boldsymbol{\omega}|^2}{2}\right]^{-1}$.

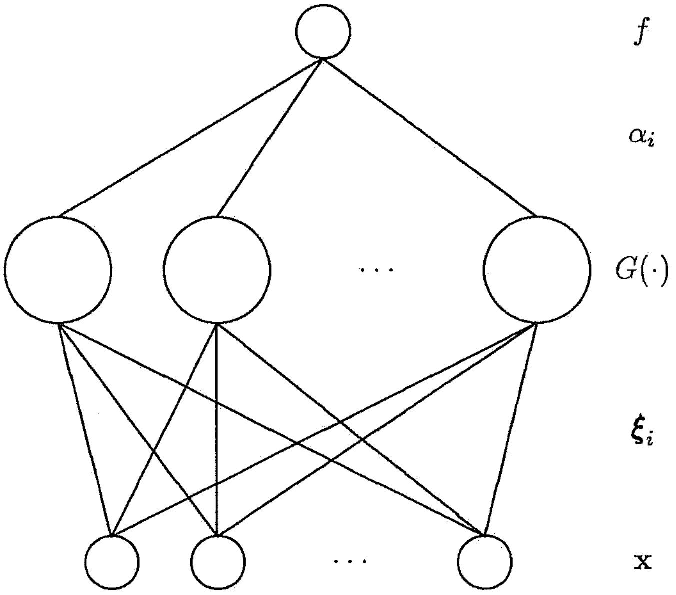

Figure 8: Radial basis function network.

The input vector $\mathbf{x}$ is subtracted from the radial basis function centers $\boldsymbol{\xi}_i$ and passed through the radial basis functions $G(\|\mathbf{x} - \boldsymbol{\xi}_i\|)$. The outputs of these functions are then summed with weights α_i to produce the output $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i G(\|\mathbf{x} - \boldsymbol{\xi}_i\|)$. The units and connections corresponding to q_k and β_k are not shown. For a vector valued output $\mathbf{f}$, the $\boldsymbol{\alpha}_i$ would also be vector valued.

centers, and weighted norms of the form

$$||\mathbf{x}||^2_{\mathbf{W}} = \mathbf{x}^T\mathbf{W}^T\mathbf{W}\mathbf{x}$$

(Poggio and Girosi, 1990). The weighting matrices $\mathbf{W}_i$, which are themselves adjusted during learning, allow the radial basis units to scale and adjust to important directions in the input space.

Since radial basis functions can be implemented as neural networks, and since they also correspond to the solutions of certain regularized problems, we can conclude that such networks provide solutions to the learning problem formulated using regularization (Poggio and Girosi, 1990). In fact, it can be shown that radial basis function networks with fixed centers have the *best approximation property* (Girosi and Poggio, 1989), i.e., given a parametrized family of radial basis function networks with a fixed number of centers, there is a single network which is the best (closest) approximation to any given continuous function. However, even better approximations can sometimes be obtained by having several competing models and allowing the data itself to dictate which model to apply (Box and Tiao, 1973; MacKay, 1992a).

Regularization has also been applied to other aspects of neural networks. A simple form of regularization is adding a weight decay term during the learning process, which can significantly speed up learning rates (Nowlan and Hinton, 1992; Nowlan, 1992; Weigend, Rumelhart and Huberman, 1991). A Bayesian choice of the weight decay parameters improves the performance even more (Nowlan and Hinton, 1992; MacKay, 1992b). When the weights in a neural network have some spatial meaning and inherent regularity or smoothness, such as the weights describing formants in a short-time periodogram, spatial regularization can be applied directly to the weight values themselves (Lang, 1989). Extracting regularities from temporal sequences using units with different time constants can be viewed as a form of temporal regularization (Mozer, 1992). The learning of spatial regularities, such as center-surround structures (Linsker, 1988) or the neighborhood connections in stereo matching (Becker and Hinton, 1989; Becker and Hinton, 1992) can be viewed as examples of regularization principles being learned *ab initio.* Finally, the appropriate amount and order of regularization to be applied, e.g., in discontinuity detection, can also be learned from examples (Aloimonos and Shulman, 1989; Becker and Hinton, 1992).

11 Discussion and Summary

As we have seen, regularization is an important component of many neural network solutions to real-world problems. It arises from two fundamentally different sources of regularity. The first source is an underlying smoothness in the continuous field being estimated, for example the (piecewise) smooth variation in depth and orientation in an image (Section 3). The application of regularization in this context leads to the development of massively parallel algorithms based on a discrete regular sampling of the field (Section 4). These algorithms can be implemented as locally connected networks in analog VLSI (Section 5).

The second application of regularization is in the learning of general nonlinear input-output mappings for which neural nets are well suited. Here the regularity is in the smooth variation of outputs in the neighborhood of input samples, which is one way to characterize the generalization properties of such networks (Section 10). In this case, however, the space in which the mapping is embedded is high dimensional (n-dimensional, where n is the number of input units), as opposed to the two-dimensional spaces typical of visual processing problems.

Underlying both of these applications is a Bayesian interpretation of regularization (Section 7), which can be used to quantify the uncertainty in the estimates produced by regularization, to develop alternative minimization algorithms, and to estimate unknown model parameters (Section 8). The Bayesian approach also permits the use of more reliable statistical techniques such as robust estimation (Section 9).

While regularization has already proven to be extremely useful in these contexts, it must be further generalized and developed to work in more real-world situations. In particular, the existence of discontinuities in smooth mappings and the presence of outliers in sampled data are both ubiquitous. In visual processing, such extensions to standard regularization theory are becoming more widely used. In learning theory, since regularization is a relatively recent addition, such extensions have yet to be explored. The addition of robust regression techniques (Huber, 1981) to learning should be relatively straightforward and should improve performance on noisy training sets (Martin and Connor, 1993). However, applying discontinuity detection, which is already a difficult problem in 2-D, to the high-dimensional spaces typical in neural networks may prove to be difficult.

Another issue that comes up very often in the context of regularization is the question of scale and of multiresolution representations. In low-level visual processing, multiresolution representations have been successfully used to speed up relaxation algorithms and to find good global solutions (Section 4). In neural network learning, multiresolution techniques have also been developed (Mjolsness, Garrett and Miranker, 1991; Moody, 1989). Finding useful multiresolution representations may be one of the keys to efficiently representing correlations and mappings in the high-dimensional spaces typical in neural networks.

Finally, while regularization has usually been applied to static problems, it has the potential to be even more useful in the context of real-time processing. For example, we can view the incremental estimation of depth fields from motion sequences (Matthies, Kanade and Szeliski, 1989) as a type of "regularization in space-time" (Szeliski, 1989). Similar concepts could be developed for neural networks which must process dynamic sequences of data or perform on-line learning and adaptation.

To summarize, regularization is a powerful technique for solving ill-posed inverse problems such as visual reconstruction and multidimensional function approximation. It provides a coherent framework for solving certain vision problems using parallel algorithms and networks, and for specifying the task of neural network learning in an architecture independent manner. It also has a rigorous justification, in term of Bayesian models, that can be used to reason about the quality of the solutions and to improve the estimation techniques. It thus forms a very useful tool for specifying, analyzing, and solving many problems related to neural network processing.

References

Aarts, E. and Korst, J. (1994). Simulated annealing. In *Mathematical Perspectives on Neural Networks.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Adler, R. J. (1981). *The Geometry of Random Fields.* J. Wiley, Chichester, England.

Ahlberg, J. H., Nilson, E. N., and Walsh, J. L. (1967). *The Theory of Splines and their Applications.* Academic Press, New York.

Aloimonos, J. and Shulman, D. (1989). *Integration of Visual Modules: An Extension of the Marr Paradigm.* Academic Press, Boston, MA.

Anderssen, R. S. and Bloomfield, P. (1974a). Numerical differentiation procedures for non-exact data. *Numerische Mathematik*, 22:157–182.

Anderssen, R. S. and Bloomfield, P. (1974b). A time series approach to numerical differentiation. *Technometrics*, 16(1):69–75.

Axelsson, O. and Barker, V. A. (1984). *Finite Element Solution of Boundary Value Problems: Theory and Computation.* Academic Press, Inc., Orlando, Florida.

Ballard, D. H. and Brown, C. M. (1982). *Computer Vision.* Prentice-Hall, Englewood Cliffs, New Jersey.

Bank, R. E., Dupont, T. F., and Yserentant, H. (1988). The hierarchical basis multigrid method. *Numerische Mathematik*, 52:427–458.

Barnard, S. T. (1989). Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3(1):17–32.

Barnard, S. T. and Fischler, M. A. (1982). Computational stereo. *Computing Surveys*, 14(4):553–572.

Barrow, H. G. and Tenenbaum, J. M. (1978). Recovering intrinsic scene characteristics from images. In Hanson, A. R. and Riseman, E. M., editors, *Computer Vision Systems*, pages 3–26. Academic Press, New York.

Barrow, H. G. and Tenenbaum, J. M. (1981). Computational vision. *Proceedings of the IEEE*, 69(5):572–595.

Bathe, K.-J. and Wilson, E. L. (1976). *Numerical Methods in Finite Element Analysis.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Becker, S. and Hinton, G. E. (1989). Spatial coherence as an internal teacher for a neural network. Technical Report CRG-TR-89-7, Connectionists Research Group, University of Toronto.

Becker, S. and Hinton, G. E. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163.

Bertero, M., Poggio, T., and Torre, V. (1988). Ill-posed problems in early vision. *Proceedings of the IEEE*, 76:869–889.

Besag, J. (1985). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, A 148.

Blake, A. and Zisserman, A. (1987). *Visual Reconstruction.* MIT Press, Cambridge, Massachusetts.

Boult, T. E. (1985a). Reproducing kernels for visual surface interpolation. Technical Report CUCS-186-85, Department of Computer Science, Columbia University.

Boult, T. E. (1985b). Visual surface interpolation: A comparison of two methods. Technical Report CUCS-189-85, Department of Computer Science, Columbia University.

Box, G. E. P. and Tiao, G. C. (1968). A Bayesian approach to some outlier problems. *Biometrika*, 55:119–129.

Box, G. E. P. and Tiao, G. C. (1973). *Bayesian Inference in Statistical Analysis.* Addison-Wesley, Reading, Massachusetts.

Brandt, A. (1977). Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, 31.

Brandt, A. (1981). Multigrid solvers on parallel computers. In Schultz, M. H., editor, *Elliptic Problem Solvers*, pages 39–83, New York. Academic Press.

Briggs, W. L. (1987). *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania.

Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355.

Carnevali, P., Coletti, L., and Patarnello, S. (1985). Image processing by simulated annealing. *IBM Journal of Research and Development*, 29(6):569–579.

Chou, P. B. and Brown, C. M. (1990). The theory and practice of Bayesian image labeling. *International Journal of Computer Vision*, 4(3):185–210.

Clark, J. J. and Yuille, A. L. (1990). *Data Fusion for Sensory Information Processing Systems.* Kluwer Academic Publishers, Boston, Massachusetts.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403.

Cybenko, G. (1994). Function approximation. In *Mathematical Perspectives on Neural Networks.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Dahlquist, G. and Björk, A. (1974). *Numerical Methods.* Prentice-Hall, Englewood Cliffs, New Jersey.

Darrell, T. and Pentland, A. (1991). Against edges:function approximation with multiple support maps. In *Advances in Neural Information Processing Systems 4*. Morgan Kauffman.

Dev, P. (1974). Segmentation processes in visual perception: A cooperative neural model. COINS Technical Report 74C-5, University of Massachusetts at Amherst.

Drumheller, M. and Poggio, T. (1986). On parallel stereo. In *IEEE International Conference on Robotics and Automation*, pages 1439–1448, San Francisco, California. IEEE Computer Society Press.

Duchon, J. (1976). Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *Revue Francaise d'Automatique, Informatique et Recherche Operationelle*, 10(12):5–12.

Duda, R. O. (1982). Kriging with derivative information. Technical Report 2, Fairchild Laboratory for Aritificial Intelligence Research.

Durbin, R., Szeliski, R., and Yuille, A. (1989). An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation*, 1(3):348–358.

Durbin, R. and Willshaw, D. (1987). An analogue approach to the traveling salesman problem using an elastic net method. *Nature*, 326:689–691.

Farin, G. E. (1992). *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, Boston, Massachusetts, 3rd edition.

Gamble, E. and Poggio, T. (1987). Visual integration and detection of discontinuities: the key role of intensity edges. A. I. Memo 970, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Geiger, D. and Girosi, F. (1991). Mean field theory for surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(5):401–412.

Geiger, D. and Yuille, A. (1991). A common framework for image segmentation. *International Journal of Computer Vision*, 6(3):227–243.

Gelb, A., editor (1974). *Applied Optimal Estimation.* MIT Press, Cambridge, Massachusetts.

Geman, D. and Hwang, C.-R. (1986). Diffusions for global optimization. *SIAM Journal of Control and Optimization*, 24(5):1031–1043.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

George, A. and Liu, J. W. H. (1981). *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Englewood Cliffs, New Jersey.

Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical Optimization.* Academic Press, Inc., London.

Girosi, F. and Poggio, T. (1989). Networks and the best approximation property. A. I. Memo 1164, Massachusetts Institute of Technology.

Grimson, W. E. L. (1981). *From Images to Surfaces: a Computational Study of the Human Early Visual System.* MIT Press, Cambridge, Massachusetts.

Grimson, W. E. L. (1983). An implementation of a computational theory of visual surface interpolation. *Computer Vision, Graphics, and Image Processing*, 22:39–69.

Hackbusch, W. and Trottenberg, U., editors (1982). *Multigrid Methods*, volume 960 of *Lecture Notes in Mathematics*, Berlin, Heidelberg, New York. Springer-Verlag.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions.* MIT Press, Cambridge, Massachusetts.

Harris, J. G. (1987). A new approach to surface reconstruction: the coupled depth/slope model. In *First International Conference on Computer Vision (ICCV'87)*, pages 277–283, London. IEEE Computer Society Press.

Harris, J. G., Koch, C., and Luo, J. (1990). A two-dimensional analog VLSI circuit for detecting discontinuities in early vision. *Science*, 248:1209–1211.

Harris, J. G., Koch, C., Staats, E., and Luo, J. (1990). Analog hardware for detecting discontinuities in early vision. *International Journal of Computer Vision*, 4(3):211–223.

Harris, J. G., Liu, S.-C., and Mathur, B. (1991). Discarding outliers in a non-linear resistive network. In *International Joint Conference on Neural Networks*, pages 501–506.

Hewett, T. A. (1986). Fractal distributions of reservoir heterogeneity and their influence on fluid transport. In *SPE 15386, 61st Annual Technical Conference and Exibition of the Society of Petroleum Engineers*, New Orleans, Louisiana. Society of Petroleum Engineers.

Hewett, T. A. (1988). Conditional simulation of reservoir heterogeneity with fractals. In *SPE 18326, 63rd Annual Technical Conference and Exibition of the Society of Petroleum Engineers*, pages 645–660, Houston, Texas. Society of Petroleum Engineers.

Hildreth, E. C. (1986). Computing the velocity field along contours. In Badler, N. I. and Tsotsos, J. K., editors, *Motion: Representation and Perception*, pages 121–127, New York. North-Holland.

Hillis, W. D. (1985). *The Connection Machine.* MIT Press, Cambridge, Massachusetts.

Hinton, G., Williams, C. K. I., and Revow, M. D. (1992). Adaptive elastic models for hand-printed character recognition. In *Advances in Neural Information Processing Systems*, volume 4, San Mateo, California. Morgan Kaufmann Publishers.

Hinton, G. E. and Nowlan, S. J. (1990). The bootstrap Widrow-Hoff rule as a cluster-formation algorithm. *Neural Computation*, 2(3):355–362.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences U.S.A.*, 81:3088–3092.

Hopfield, J. J. and Tank, D. W. (1985). 'Neural' computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152.

Horn, B. K. P. (1974). Determining lightness from an image. *Computer Graphics and Image Processing*, 3(4):277–299.

Horn, B. K. P. (1977). Understanding image intensities. *Artificial Intelligence*, 8(2):201–231.

Horn, B. K. P. (1986). *Robot Vision.* MIT Press, Cambridge, Massachusetts.

Horn, B. K. P. (1988). Parallel networks for machine vision. A. I. Memo 1071, Massachusetts Institute of Technology.

Horn, B. K. P. (1990). Height and gradient from shading. *International Journal of Computer Vision*, 5(1):37–75.

Horn, B. K. P. and Brooks, M. J. (1989). *Shape from Shading.* MIT Press, Cambridge, Massachusetts.

Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Huber, P. J. (1981). *Robust Statistics.* John Wiley & Sons, New York.

Hummel, R. A. and Zucker, S. W. (1983). On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:267–287.

Hutchinson, J., Koch, C., Luo, J., and Mead, C. (1988). Computing motion using analog and binary resistive networks. *Computer*, 21(3):52–63.

Ikeuchi, K. and Horn, B. K. P. (1981). Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Keren, D. and Werman, M. (1990). Variations on regularization. In *10th International Conference on Pattern Recognition (ICPR'90)*, pages 93–98, Atlantic City, New Jersey. IEEE Computer Society Press.

Kimeldorf, G. and Wahba, G. (1970). A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502.

Kirkpatrick, S., Gelatt, C. D. J., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Koch, C. (1989). Seeing chips: analog VLSI circuits for computer vision. *Neural Computation*, 1(2):184–200.

Koch, C., Marroquin, J., and Yuille, A. (1986). Analog "neuronal" networks in early vision. *Proceedings of the National Academy of Sciences U.S.A.*, 83:4263–4267.

Lang, K. (1989). *A Time-Delay Neural Network Architecture for Speech Recognition.* PhD thesis, Carnegie Mellon University.

Laurent, P. J. (1972). *Approximation et Optimisation.* Hermann, Paris.

Lehky, S. R. and Sejnowski, T. J. (1988). Network model of shape-from-shading: neural function arises from both receptive and projective fields. *Nature*, 333:452–454.

Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, 21(3):105–117.

Little, J., Blelloch, G. E., and Cass, T. A. (1989). Algorithmic techniques for computer vision on a fine-grained parallel machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(3):244–257.

Little, J., Bulthoff, H., and Poggio, T. (1987). Parallel optical flow computation. In *Image Understanding Workshop*, pages 915–920, Los Angeles, California. Morgan Kaufmann Publishers.

MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

MacKay, D. J. C. (1992b). A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472.

Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature.* W. H. Freeman, San Francisco, California.

Marr, D. (1978). Representing visual information. In Hanson, A. R. and Riseman, E. M., editors, *Computer Vision Systems*, pages 61–80. Academic Press, New York.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman, San Francisco, California.

Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, 194:283–287.

Marroquin, J., Mitter, S., and Poggio, T. (1987). Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82(397):76–89.

Martin, R. D. and Connor, J. (1993). Robust neural networks for regression and time series prediction. In *Neural Networks for Computing*, Snowbird, Utah.

Matthies, L. H., Kanade, T., and Szeliski, R. (1989). Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236.

Mead, C. (1989). *Analog VLSI and Neural Systems.* Addison-Wesley, Reading, Massachusetts.

Mead, C. A. and Mahowald, M. A. (1988). A silicon model of early visual processing. *Neural Networks*, 1:91–97.

Meinguet, J. (1979). Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics (ZAMP)*, 30:292–304.

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

Micchelli, C. A. (1986). Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22.

Mjolsness, E., Garrett, C. D., and Miranker, W. L. (1991). Multiscale optimization in neural nets. *IEEE Transactions on Neural Networks*, 2(2):263–274.

Mjolsness, E., Gindi, G., and Anandan, P. (1989). Optimization in model matching and perceptual organization. *Neural Computation*, 1(2):218–229.

Moody, J. (1989). Fast learning in multi-resolution hierarchies. In Touretzky, D., editor, *Advances in Neural Information Processing Systems*, volume 1, pages 29–39. Morgan Kaufmann Publishers, Carnegie Mellon University.

Moody, J. and Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294.

Morozov, V. A. (1984). *Methods for Solving Incorrectly Posed Problems.* Springer-Verlag, New York.

Mozer, M. C. (1992). The induction of multiscale temporal structure. In *Advances in Neural Information Processing Systems IV*, pages 275–282, San Mateo, California. Morgan Kaufmann Publishers.

Mozer, M. C., Zemel, R. S., Behrmann, M., and Williams, C. K. I. (1992). Learning to segment images using dynamic feature binding. *Neural Computation*, 4:650–665.

Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 52:577–685.

Nagel, H.-H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5):565–593.

Nowlan, S. J. (1992). *Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures.* PhD thesis, Carnegie Mellon University.

Nowlan, S. J. and Hinton, G. E. (1992). Adaptive soft weight tying using Gaussian mixtures. In *Advances in Neural Information Processing Systems*, volume 4, San Mateo, California. Morgan Kaufmann Publishers.

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical Science*, 1(4):502–527.

Parisi, G. (1988). *Statistical Field Theory.* Addison-Wesley, Reading, Massachusetts.

Pentland, A. (1989a). Part segmentation for object recognition. *Neural Computation*, 1(1):82–91.

Pentland, A. (1989b). A possible neural mechanism for computing shape from shading. *Neural Computation*, 1(2):208–217.

Perona, P. and Malik, J. (1990). Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.

Platt, J. (1991). A resource allocating network for function interpolation. *Neural Computation*, 3(1):213–225.

Poggio, T. and Edelman, S. (1990). A network that learns to recognize 3D objects. *Nature*, 343:263–266.

Poggio, T., Gamble, E. B., and Little, J. J. (1988). Parallel integration of vision modules. *Science*, 242:436–440.

Poggio, T. and Girosi, F. (1989). A theory of networks for approximation and learning. A. I. Memo 1140, Massachusetts Institute of Technology.

Poggio, T. and Girosi, F. (1990). A theory of networks for approximation and learning. *Science*, 247:978–982.

Poggio, T. and Koch, C. (1985). Ill-posed problems in early vision: from computational theory to analogue networks. *Proceedings of the Royal Society of London*, B 226:303–323.

Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317(6035):314–319.

Poggio, T., Voorhees, H., and Yuille, A. (1985). A regularized solution to edge detection. A. I. Memo 833, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Potter, J. L., editor (1985). *The Massively Parallel Processor.* MIT Press, Cambridge, Massachusetts.

Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review. In *Algorithms for Approximation*, pages 143–167. Clarendon Press, Oxford.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1992). *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, England, second edition.

Reinsch, G. (1967). Smoothing by spline functions. *Numerische Mathematik*, 10:177–183.

Rosenfeld, A., editor (1984). *Multiresolution Image Processing and Analysis*, New York. Springer-Verlag.

Rumelhart, D. (1994). Probabilistic interpretation. In *Mathematical Perspectives on Neural Networks.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., and the PDP research group, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*, volume I. Bradford Books, Cambridge, Massachusetts.

Schumaker, L. L. (1981). *Spline Functions: Basic Theory.* John Wiley & Sons, New York.

Sinha, S. S. and Schunck, B. G. (1990). A robust method for surface reconstruction. In *IEEE Workshop on Robust Techniques in Computer Vision*, pages 183–199, Seattle, Washington. IEEE Computer Society Press.

Suter, D. (1991). Mixed finite element based neural networks in vision. *International Journal of Pattern Recognition and Artificial Intelligence.*

Suter, D. and Mansor, D. (1991). Regularization and spline fitting by analog neural networks. *IEEE Transactions on Neural Networks.*

Szeliski, R. (1987). Regularization uses fractal priors. In *Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 749–754, Seattle, Washington. Morgan Kaufmann Publishers.

Szeliski, R. (1989). *Bayesian Modeling of Uncertainty in Low-Level Vision.* Kluwer Academic Publishers, Boston, Massachusetts.

Szeliski, R. (1990). Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528.

Szeliski, R. (1991). Fast shape from shading. *CVGIP: Image Understanding*, 53(2):129–153.

Szeliski, R. and Hinton, G. (1985). Solving random-dot stereograms using the heat equation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'85)*, pages 284–288, San Francisco, California. IEEE Computer Society Press.

Terzopoulos, D. (1984). *Multiresolution Computation of Visible-Surface Representations.* PhD thesis, Massachusetts Institute of Technology.

Terzopoulos, D. (1986a). Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):129–139.

Terzopoulos, D. (1986b). Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):413–424.

Terzopoulos, D. (1988). The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):417–438.

Terzopoulos, D., Witkin, A., and Kass, M. (1988). Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123.

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of Ill-Posed Problems.* V. H. Winston, Washington, D. C.

Ullman, S. (1979). Relaxation and constrained optimization by local processes. *Computer Graphics and Image Processing*, 10:115–125.

Wahba, G. (1983). Bayesian "confidence intervals" for the cross-validated smoothing spline. *Journal of the Royal Statistical Society*, B 45(1):133–150.

Wahba, G. and Wendelberger, J. (1980). Some new mathematical methods for variational objective analysis using splines and cross validation. *Monthly Weather Review*, 108:1122–1143.

Waltz, D. L. (1975). Understanding line drawings of scenes with shadows. In Winston, P., editor, *The Psychology of Computer Vision*. McGraw-Hill, New York.

Watrous, R. (1994). Numerical optimization. In *Mathematical perspectives on neural networks*, Developments in Connectionist Theory. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Wechsler, H. (1990). *Computational Vision*. Academic Press, San Diego, California.

Weigend, A. S., Rumelhart, D. E., and Huberman, B. A. (1991). Generalization by weight-elimination with applications to forecasting. In *Advances in Neural Information Processing Systems*, volume 3, pages 875–882, Palo Alto, California. Morgan Kaufmann Publishers.

Witkin, A., Terzopoulos, D., and Kass, M. (1987). Signal matching through scale space. *International Journal of Computer Vision*, 1:133–144.

Wolpert, D. H. (1993). On the use of evidence in neural networks. In *Advances in Neural Information Processing Systems*, volume 5, San Mateo, California. Morgan Kaufmann Publishers.

Yserentant, H. (1986). On the multi-level splitting of finite element spaces. *Numerische Mathematik*, 49:379–412.

Yuille, A. L. (1990). Generalized deformable models, statistical physics, and matching problems. *Neural Computation*, 2(1):1–24.

Yuille, A. L. and Grzywacz, N. M. (1988). A computational theory for the perception of coherent visual motion. *Nature*, 333:71–74.

Zienkiewicz, O. C. (1972). *The Finite Element Method.* McGraw-Hill, New York, 3 edition.

Zucker, S. W., Dobbins, A., and Iverson, L. (1989). Two stages of curve detection suggest two styles of visual computation. *Neural Computation*, 1(1):68–81.