MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

SOLVING INVERSE PROBLEMS IN COMPUTER VISION BY SCALE SPACE RECONSTRUCTION

Alun Jones and Christopher Taylor

Department of Medical Biophysics, University of Manchester, Manchester, M13 9PT, U.K.
Phone: +44 061 275 5130, Fax: +44 061 275 5145. Email: agj@wiau.mb.man.ac.uk

ABSTRACT

Ill–posed inverse problems are widely encountered in computer vision, examples include shape from shading, surface reconstruction from sparse data and optic flow. Unique solutions to these problems are conventionally found by minimizing an objective function regularized by a smoothness constraint. However, objective functions of this form often contain many local minima, making it difficult to find an adequate solution by standard numerical methods.

We describe an algorithm for solving inverse problems using scale space tracking which is robust, provably convergent and avoids local minima. The algorithm generates a hierarchy of solutions at different scales, forming a scale space from which a final solution can be selected at a later stage. We show how the use of gaussian basis functions to construct solutions can result in scale space behaviour without the need to blur the input data.

Results are shown for shape from shading and surface reconstruction from stereo data, using both real and synthetic images.

INTRODUCTION

Almost every important problem encountered in computer vision may be viewed as an ill–posed inverse problem: inverse because we are taking the end result of the physical process of imaging and wish to deduce something about the observed scene, and ill–posed because the imaging process discards much of the information in the scene being observed. To obtain solutions to these problems we apply (either wittingly or unwittingly) heuristic assumptions about the solution we expect, and much of current computer vision research is involved in finding useful assumptions to make.

The classical approach to solving ill–posed inverse problems found in the numerical analysis literature involves minimizing an objective function regularized with a smoothness constraint, typically of the following form:

$$\mathcal{A}(\mathbf{u}) + \lambda\mathcal{B}(\mathbf{u}) \qquad (1)$$

where $\mathcal{A}$ is a measure of how well the solution **u** fits the data, $\mathcal{B}$ is a "lack–of–smoothness" measure (eg the sum of squared second derivatives), and λ controls the trade–off between explaining the data and smoothness. The heuristic assumption being made here is that the solution should be smooth in some sense, since most functions encountered when modeling the real world are smooth. This assumption has been found to be valid in many areas of computer vision: for example, the fact that physical surfaces tend to be smooth can be used in surface reconstruction from sparse data and shape from shading; and the fact that most moving objects are rigid and travel in a smooth trajectory allows smoothness to be used in optic flow and motion problems.

Whilst the regularization approach has been used successfully in a wide range of domains, the nature of computer vision problems has highlighted several drawbacks:

- When the objective function contains local minima standard optimization algorithms tend to get stuck in them and are unable to reach an adequate solution.
- The large–scale nature of vision problems mean simple iterative schemes may need to be used (via, for example, the calculus of variations), with convergence not guaranteed.
- The value of the constant λ is difficult to choose objectively and is usually found by guess-work (and is therefore sub–optimal).

The algorithm described in this paper solves inverse problems using *scale space tracking* [1] and is robust, provably convergent and does not require prior decisions to be made about the smoothness of the solution. Whilst our method is not in general guaranteed to find the optimal solution, it will usually find a "significant" solution which is acceptable for many problems [2].

The algorithm works by generating a hierarchy of solutions at different scales, forming a *scale space* [3] from which an appropriately smooth final solution can be selected at a later stage. Two heuristic assumptions are made about the solution: that it is smooth (although we do not need to make a prior decision about exactly how smooth it should be); and that it is continuous. These assumptions are appropriate for a wide class of vision problems including shape from shading, stereo matching, surface–fitting to sparse data and image registration.

We have applied the technique to shape from shading and to surface reconstruction from stereo data, achieving promising results in both cases. We believe our approach offers an effective alternative to regularization for many of the large–scale inverse problems found in computer vision.

SCALE SPACE

It has long been realised that an image possesses structure at a range of scales and that having a description of the image that highlights the activity at each scale is of great use in analyzing the image (eg [3]). A method for decomposing an image (or more generally, a signal) according to scale was first proposed by Witkin [1] which involves progressively blurring the image with a gaussian to form the *scale space*, *L(x,y;t)*:

$$L(x,y;t) = \int_{u=-\infty}^{\infty}\int_{v=-\infty}^{\infty} \frac{1}{2\pi t} e^{\frac{-(u^2+v^2)}{2t}} I(x-u,y-v)\,dv\,du \qquad (2)$$

where $I(x,y)$ is the image and $t > 0$ is the scale parameter. The convolution integral (2) is also the solution to the two-dimensional diffusion equation:

$$\frac{\partial L}{\partial t} = \frac{1}{2}\left(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}\right) \qquad (3)$$

with the initial condition $L(x,y;0) = I(x,y)$. A discrete scale space can therefore be formed either by discretizing (1) and performing the convolutions, or by discretizing (3) in space with the five– or nine–point Laplace operator and then solving the resultant system of coupled ordinary differential equations using a standard numerical method such as the θ –method.

SCALE SPACE TRACKING

The concept of scale space has proved useful in minimization techniques for solving the large–scale non–convex problems which commonly occur in computer vision. The idea is to first find an approximate solution to the problem at a very large scale and then "track" this solution as the scale is gradually reduced [1,2]. At an appropriately large starting scale the problem will be convex so the first approximation to the solution can easily be found, and although local minima reappear as the scale is reduced, the fact that the solution is being tracked through scale ensures that the nearest minimum is the one required.

To formalise this procedure, the problem must first be expressed as the minimization of an energy functional, $E(\mathbf{u})$, where $\mathbf{u}$ is the solution collapsed into a vector. At each scale it is required that the energy function is at a minimum, so we have the condition $\nabla E(\mathbf{u}) = 0$. To maintain this equilibrium, it is necessary to minimize E at each new scale, either by using an optimization technique or by solving $\nabla E(\mathbf{u}) = 0$ directly. The discrete transition from one scale to the next is most simply done by taking the solution at the previous scale as the initial condition for the minimization of E at the new scale. We can then write the scale space tracking algorithm as the solution of the recurrence relation:

$$\mathbf{u}(t_k) = \mathbf{u}(t_{k+1}) + \psi(t_k) \qquad k = N, N-1, \ldots, 0 \qquad (4)$$

where $\psi(t_k)$ is a deformation of the solution such that $E(\mathbf{u}(t_k) + \psi(t_{k+1}))$ is at a minimum, and the initial condition is $\mathbf{u}(t_N) = \mathbf{0}$. More advanced algorithms project the solution from one scale to the next along the tangent to the scale space trajectory [4], but this approach involves the calculation of the Hessian matrix of the energy function $H = \nabla\nabla E$ which is of high dimension (although it may be sparse for certain formulations [4]).

It should be stressed that it is the *solution* that is being tracked through scale space, so we require that scale space behaviour is imposed on the solution space. In the early work on scale space tracking (eg [1,2]) it was the *input data* (typically image data) that was gaussian blurred to form the scale space. For those problems where the relationship between the input data and the solution is linear this approach leads to the solution space also having scale space behaviour, but in those cases where a non–linear relationship exists the solution will not be correctly blurred. An example where blurring the input data is not appropriate occurs in shape from shading, where the relationship between the image $I(x,y)$ and the surface $z(x,y)$ is given by the image irradiance equation $I(x,y) = R(z(x,y))$. The reflectance function R is in general non–linear and will therefore not commute with the blurring operator B, ie $B(R(z(x,y))) \neq R(B(z(x,y)))$ [5].

SCALE SPACE RECONSTRUCTION

Since blurring the input data only leads to correct scale space behaviour in the solution space if the relationship between the input data and the solution is linear, we instead choose to impose scale space behaviour onto the solution directly. We introduce the term "scale space reconstruction" for algorithms that work in this way, since they must start without any scale space information and gradually reconstruct the entire solution scale space. In effect, this means solving differential equation (3) "in reverse" (t decreasing) with the initial condition $L(x,y;\infty) = 0$, making use of the fact that the energy function E must be minimized as the algorithm progresses.

This approach was first adopted by Whitten [4] using solutions constructed from the deformable curves, or "snakes", of Kass *et al.* [6]. Whitten realized that the concepts of smoothness and scale are essentially the same, so by controlling the smoothness (internal energy) of the snakes, scale space behaviour could be achieved. The rest of this paper describes an alternative approach that drops the scale dependent energy functional of Whitten, $E(\mathbf{u}(t),t)$, in favour of a simpler objective function $F(\mathbf{u})$ that does not depend on scale. Scale space behaviour is not enforced via the energy functional, but by the tracking algorithm itself and the choice of basis functions for the solution.

We will use the equivalence between scale and smoothness in the following discussion. First, let us consider the starting situation where we have no more information about the solution than the initial approximation $\mathbf{u} = \mathbf{0}$. We can make a small change to $\mathbf{u}$ that decreases the objective function, and this will be accompanied with a loss of smoothness (corresponding to a decrease in scale). We would wish to choose a direction for the change that causes the greatest decrease in the objective function for the least cost in smoothness, ie the direction which maximizes the ratio:

$$-\frac{F(\mathbf{u} + \delta\mathbf{u}) - F(\mathbf{u})}{S(\mathbf{u} + \delta\mathbf{u}) - S(\mathbf{u})} = \frac{-\Delta F}{\Delta S} \qquad (5)$$

where $\delta\mathbf{u}$ is the change in the solution, and $S(\mathbf{u})$ is a measure of the lack–of–smoothness (or "roughness") of the solution (eg the sum of squared second derivatives) so ΔS corresponds to a reduction in scale. If $\delta\mathbf{u}$ is made infinitesimal, this direction can be approximated by the vector:

$$\mathbf{s} = \left[\frac{-\partial F/\partial u_1}{e^{(\partial S/\partial u_1)}}, \ldots, \frac{-\partial F/\partial u_n}{e^{(\partial S/\partial u_n)}}\right]^T \qquad (6)$$

Moving the solution a small distance in the direction $\mathbf{s}$ gives a new point where a new direction can be calculated using (6) and the trajectory continued, and so on. Unfortunately, such an approach is likely to require very small step lengths and would be very badly conditioned.

The conditioning of the problem can be improved by parameterizing the deformation of the solution using gaussian basis functions of width appropriate for the current scale. The gaussian is an suitable function since it is the impulse response of the gaussian blurring kernel normally used to form a scale space by decomposition. We can now write the solution at a particular scale as:

$$\mathbf{u}_k = \mathbf{u}_{k+1} + \mathbf{Ga} \qquad (7)$$

where $\mathbf{G}$ is the gaussian convolution matrix $[g_{i,j}]$, $g_{i,j} = e^{-|\mathbf{x}_i - \mathbf{x}_j|^2/2\sigma_k}$, σ_k controls the width of the gaussian, $\mathbf{a}$ is the vector of gaussian amplitude coefficients and $\mathbf{x}_i$ is the grid co-ordinate of the *i*th gaussian. The initial condition for (7) is $\mathbf{u}_N = \mathbf{0}$, and the recurrence relation is run until the full resolution solution $\mathbf{u}_0$ is reached. The width of the gaussian basis functions is reduced in stages as the scale decreases, so $\sigma_k < \sigma_{k+1}$ for all k.

During each iteration of (7), the trajectory of the solution is advanced using the maximum downhill principle of (5), except that now it is the gaussian coefficients **a** that are adjusted rather than the solution **u**. After continuing the trajectory in this way for a while, progress becomes difficult as the conditioning of the problem worsens due to the current gaussian basis functions no longer being appropriate at the reduced scale. The iteration must then be completed and a new set of slightly narrower gaussian basis functions adopted ready for the next iteration.

To improve the computational efficiency of the implementation we relax the maximum downhill requirement of (5) and instead use the standard numerical technique of *conjugate gradient descent* [7] to minimize the objective function F without regard to changes in smoothness. We can ignore changes in smoothness since the use of gaussian basis functions puts a bound on the loss of smoothness, ensuring that taking reasonably small steps in **a** along *any* direction does not introduce large losses of smoothness. It was shown in [8] that a bound on the lack–of–smoothness for the solution $\mathbf{u}_k$ is given by: $S(\mathbf{u}_k) \leq S(\mathbf{u}_{k+1}) + C_k \|\mathbf{a}\|^2$, where C_k is a constant and $C_k > C_{k+1}$. Since the bound is dependent on C_k, which increases as the gaussians become narrower, a change of basis to narrower gaussians should not be made until the rate of convergence using the current basis is so low as to make the change necessary.

The set of solutions $\mathbf{u}_k$, $k = 0,...,N$ define a scale space from which the final solution can be selected. For a least-squares objective function this may be done by choosing the smoothest solution where the value of chi-squared is less than the number of independent measurements. We are investigating other criteria for selecting the final solution.

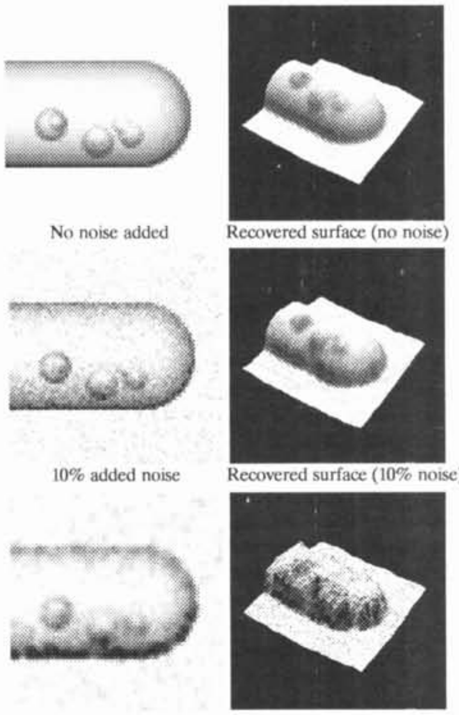
EXAMPLES

For shape from shading an appropriate objective function is the brightness error between the observed image and the rendered image of the solution surface [8]:

$$F = \sum_{x,y} (I(x,y) - R(p,q))^2$$

where $I(x,y)$ is the image, R is the reflectance function, $p = \partial z(\mathbf{u})/\partial x$, $q = \partial z(\mathbf{u})/\partial y$ and $z(\mathbf{u})$ is the surface.

We show results for a synthetic image with Lambertian shading for various amounts of added gaussian noise (Figure 1). Figure 2 shows samples from scale space for the noiseless synthetic image. Recovered surfaces are also shown for real scanning electron microscope (SEM) images of a cylindrical fibre and an SEM sample grid (Figure 3).



No noise added

Recovered surface (no noise)

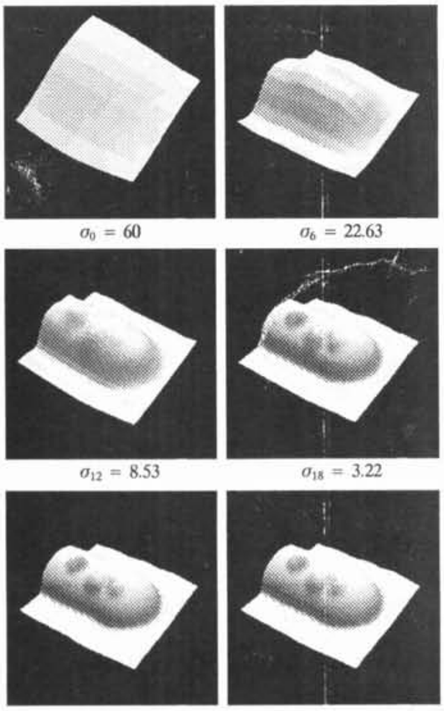
10% added noise

Recovered surface (10% noise)

20% added noise

Recovered surface (20% noise)

Figure 1: Recovered surfaces for synthetic images



$\sigma_0 = 60$

$\sigma_6 = 22.63$

$\sigma_{12} = 8.53$

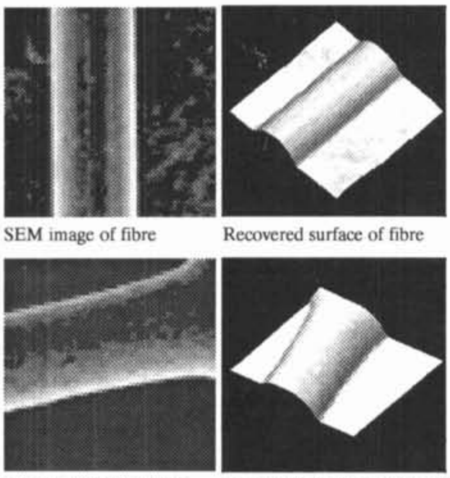
$\sigma_{18} = 3.22$

$\sigma_{24} = 1.21$

$\sigma_{31} = 0.39$

Figure 2: Samples from scale space

In the stereo surface reconstruction problem, the best match is first found for each surface point using 3-by-3 template matching, after which probability estimates for these



SEM image of fibre Recovered surface of fibre

SEM image of EM grid Recovered surface of EM grid

Figure 3: SFS recovered surfaces for SEM images

matches are found by applying Bayes' rule. This leads to the maximum likelihood estimator for the surface:

$$F = \sum_{x,y} \frac{(D(\Phi(x,y;\sigma)) - d(x,y))^2}{s(x,y)^2}$$

where D takes the height of a surface point and calculates its disparity, $d(x,y)$ is the disparity of the best match for the surface point, and $s(x,y)$ is a confidence estimate for the match. Figure 4 shows a stereo pair of SEM images of cylindrical fibres, and Figure 5 shows the surface recovered by the stereo algorithm.

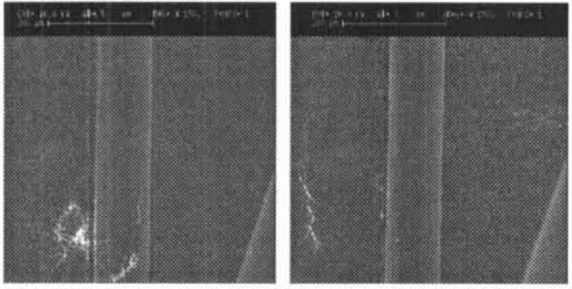


Figure 4: Stereo pair of SEM images of cylindrical fibre

SUMMARY

We have described a general method for reconstructing the scale space representation of a solution to an ill-posed inverse problem. The method builds the scale space from gaussian basis functions, which allows smoothness to be controlled in a natural way without adding extra terms to the objective function. The algorithm first finds an approximate solution at a large scale which it then tracks through scale space by making small deformations that reduce the objective function with minimum loss of smoothness. No assumptions are made about the form of the objective function and no blurring of the input data is required.

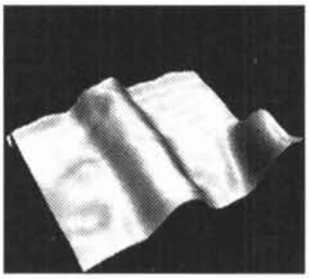


Figure 5: Recovered surface

ACKNOWLEDGEMENTS

The author holds a Total Technology studentship funded by the EPSRC and Leica Cambridge Ltd.

REFERENCES

[1] Witkin, AP. "Scale space filtering". 8th Int. Joint Conf. AI proceedings, pp1019–1022, 1983.

[2] Witkin, A, Terzopoulos, D and Kass, M. "Signal matching through scale space". Fifth Conf. AI proceedings pp714–719, 1986.

[3] Koenderinck J.J. *Solid Shape*. The MIT Press, Cambridge, MA. 1990.

[4] Whitten, G. "Scale space tracking and deformable sheet models for computational vision". *IEEE PAMI* vol 15 no 7, July 1993.

[5] Peleg S, Ron G. "Nonlinear Multiresolution: a Shape from Shading Example". *IEEE PAMI* Vol 12 num 12, Dec 1990.

[6] Kass M, Witkin A, Terzopoulos D. "Snakes: Active Contour Models". *International Journal of Computer Vision*, 1321-331, 1987.

[7] Gill PE, Murray W. *Practical optimization*. Academic Press, New York, 1981.

[8] Jones AG, Taylor CJ. "Robust Shape From Shading". *Image and Vision Computing*, September 1994.