Solving Random-Dot Stereograms Using the Heat Equation

Richard Szeliski and Geoffrey Hinton

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

Many parallel algorithms have been proposed for finding the correct matches between feature points in random dot stereograms. Some algorithms have used local support functions and have achieved globally good solutions by using relaxation in a parallel network. Recently, Prazdny[1] has shown that iteration is unnecessary if a much larger support function is used, and that this support function can be designed to work for stereograms containing transparent surfaces. We describe a simple global support function that can be efficiently implemented by relaxation in a network with only local connectivity. This function, which is the solution to the heat diffusion equation, does not work as well as Prazdny's. By using the *difference* of two heat equations, we can improve the performance and get results almost identical to Prazdny's, at a lower computational cost.

1. Introduction

Random-dot stereograms have long been used in vision to study the process of binocular depth perception. Such stereograms (image pairs) are formed by placing black dots randomly on a white background, and shifting them differentially in the left and right images. The simple form of the feature points focuses attention on the selection of the correct matches among all the possible correspondences.

This paper compares a number of stereo correspondence algorithms that use global support functions. In such algorithms, each candidate match receives votes from its neighboring matches (with similar disparity matches voting more strongly). The match with the greatest support in each ocular column is selected as the correct match. Such a scheme can be used to solve both opaque and transparent surface stereograms.

An alternative to the direct calculation of the global support function is its approximation by a heat diffusion equation, allowing analysis by the finite element method. This yields a local iterative method that is computationally more efficient, while maintaining a similar level of performance compared to the direct implementation.

The paper is divided as follows. A review of previous work in the field is presented, including the recent work on which this paper is based. The heat equation is introduced, and a finite element analysis is used to obtain an iterative implementation. A multi-resolution version is briefly discussed, followed by a discussion of the shape of the support function, and how it can be improved. Finally comparative results are presented on some sample stereograms, and the conclusions are summarized.

2. Previous work

The first investigations into random-dot stereograms began with Julesz' work[2] in which he studied stereo image pairs consisting only of random dots. These stereograms contain no monocular depth cues, and yet still yield a vivid depth perception when fused (Figure 1). In addition to generating the stereograms, Julesz also proposed a simple "spring" model that could solve the correspondence problem[3]. Several related models were subsequently proposed by Sperling[4], Dev[5], Marr and Poggio[6] and most recently Prazdny[1].

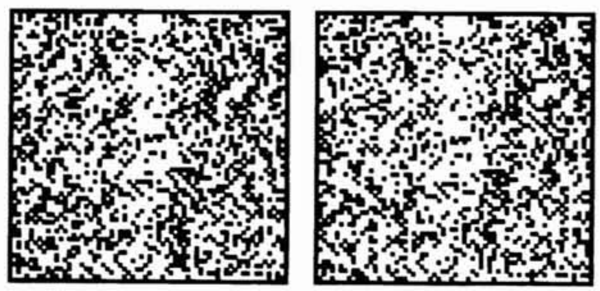


Figure 1: A random-dot stereogram showing transparent surfaces The surfaces form a series of steps with a central 0 disparity plane The dot density is 20%

The Marr-Poggio cooperative stereo algorithm is based on three rules: *compatibility*, *uniqueness* and *continuity*. By restricting our attention to random-dot stereograms, we can by-pass the difficult issue of compatibility or feature matching. Much work has been done in this field[7, 8], as well as in the development of non-parallel algorithms[9, 10]. Neither of these topics is addressed here.

Using the Marr-Poggio algorithm, the correspondence problem is solved by iteratively updating a network of linear threshold units, each of which is connected to a small number of neighbors. To implement the uniqueness rule that each point in one image usually matches only one point in the other image, matches in the same ocular column inhibit each other. To implement the continuity rule that adjacent points on a surface are at similar depths, nearby matches at the same disparity support each other. After iterating, the network settles to a solution that consists of a dense map of disparity values. In the Marr-Poggio model, the interpolation of disparity values to areas where no matches exist is necessary to achieve global support with only local interactions.

In CVPR'85, San Francisco, June 1985, pp. 284-288

Recently, Prazdny has proposed a local *non-iterative* parallel algorithm that replaces the continuity rule with a *coherence principle*[1]. This principle requires matched points to lie on some coherent (relatively smooth) surface, but does *not* require neighboring matches to be on the same surface. This principle allows the solution of transparent random-dot stereograms (Figure 1), but it cannot be implemented in a network (such as Marr-Poggio model[6]) where non-neighboring[1] matches can only support each other by interpolating phantom matches (i.e. matches where no feature points exist).

Prazdny's algorithm works by determining for each possible match a single "disparity cell" value that is the sum of the similarity support received from its neighbors. The support function is based on the *disparity gradient*, which is the slant of the line through the two points, and its strength is inversely proportional to the image plane distance between the points. Within each disparity column, the match with the highest support is selected. This process is done separately for each eye, and the results are combined. An alternative support function that uses some iteration and in addition has a fixed disparity gradient limit has also been proposed by Mayhew[11].

The algorithm presented in this paper is similar to Prazdny's, except that it replaces the global support function with one that can be calculated by iteration of a local finite difference equation. Thus, a tradeoff can be made between the number of connections and the number of iterations.

3. The heat equation

The support function that we are searching for should be analytic, amenable to finite element analysis and realizable using relaxation. It must obey superposition, to allow a strictly local implementation, and isotropic (for simplicity). The effects of one candidate match on another should fall off as the distance between the two increases.

One possible candidate for such a function might be the heat equation

$$\frac{\partial u}{\partial t} = \nabla^2 u \quad \text{with} \quad u(x,0) = f(x)$$

in which the input data is used as the starting condition[12]. Unfortunately, the blurring in this model increases with time, and the solution converges to a single uniform temperature, independent of the input. To get a useful solution we need heat sources embedded in the three-dimensional (x,y,d) space. The heat diffusion equation for such a system at equilibrium is

$$\frac{\partial u}{\partial t} = \nabla^2 u - \alpha u + p = 0$$

where p is the function defining the point sources (i.e. p is the compatibility function). The αu (decay) term ensures that the solution falls away to 0 away from the sources. Note that the equation above is similar to the *membrane* model for surface interpolation[13]

$$\nabla^2 u + \beta(p - u) = 0$$

which, however, does not fall off to 0, and does not obey superposition.

---

[1]Non-neighboring matches are those that are more widely separated than the radius of the support function

Since superposition holds, to find the total analytical solution, it is only necessary to find the solution for a single point source at the origin (see Appendix A). The solution is of the form

$$u(r) = \frac{1}{r} e^{-\sqrt{\alpha}\, r} \quad \text{(where } r \text{ is the distance from the origin)}$$

4. Finite element solution

To calculate the values of the solution of the heat equation in a local iterative fashion, a finite element approach is used. Since the differential equation to be solved is Laplace's equation, a simple linear conforming element can be used[14, 13]. With a rectangular tessellation, we can express the gradient as

$$\nabla^2 u_{x,y,d} = \kappa_1 \{u_{x+1,y,d} + u_{x-1,y,d} + u_{x,y+1,d} + u_{x,y-1,d}\} + \{u_{x,y,d+1} + u_{x,y,d-1}\} - (2 + 4\kappa_1) u_{x,y,d}$$

where κ_1 is the ratio of horizontal to vertical thermal conductivity coefficients. The heat equation can then be solved using the following iterative algorithm.

$$u^{t+\Delta t} \leftarrow u^t + \Delta t \{\nabla^2 u - \alpha u + p\}$$

where $p = 1$ at any candidate match.

Simulations of this algorithm show that it converges to the expected analytic function, except at the origin, where the function is undefined. Convergence is much more rapid for larger α's (faster decay constants). Also, the points near the origin converge rapidly to their final values. The details of the implementation are given in Appendix B.

The time required for a point to reach within a fixed percentage of its final value is linearly related to its distance from the origin (i.e. the heat source). Thus, the time complexity of the iterative algorithm is $\mathbf{O}(n)$ as opposed to Prazdny's non-iterative ($\mathbf{O}(1)$) algorithm. However, the iterative version requires only six connections ($\mathbf{O}(1)$), whereas Prazdny's requires $\mathbf{O}(n^2)$ (assuming a constant disparity limit). These are not formal complexity bounds, but are meant to suggest the tradeoff available between iteration and connection complexity.

5. Multi-resolution algorithms

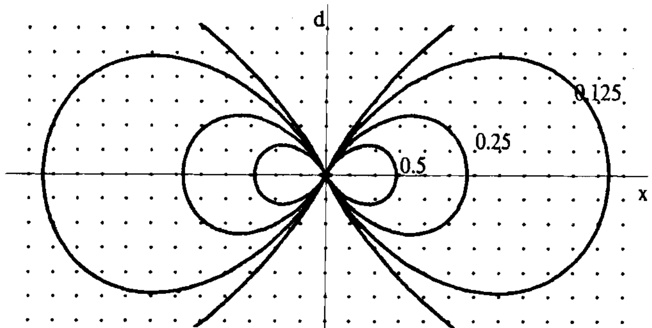
A multi-resolution implementation of the algorithm can be used to significantly speed up its convergence. This is especially true for the heat diffusion equation, since the effective α increases by 4 each time the image is subsampled by 2 (replacing r by $r/2$ in $u(r)$ yields an identical function if α is replaced by 4α).

To implement a multi-resolution version, a simple coarse-to-fine strategy should be sufficient. The same diffusion equation (and hence updating rule) is used, except that α must be scaled up, and the p's must be calculated by local smoothing. If multi-resolution matching data is available, it can be used directly as input to the various levels.

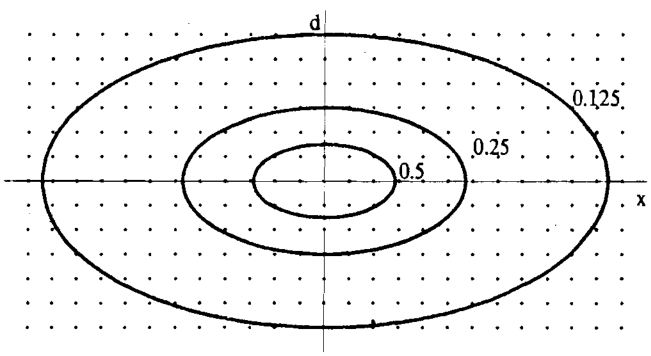
Since experimental results show that only a few (e.g. 6) iterations at the finest level are needed to solve the stereo correspondence problem, the multi-resolution version of the heat equation has not been implemented. Multi-resolution techniques would be of interest if the radius of the support function were larger, or the number of iterations had to be reduced.

6. The shape of the support function

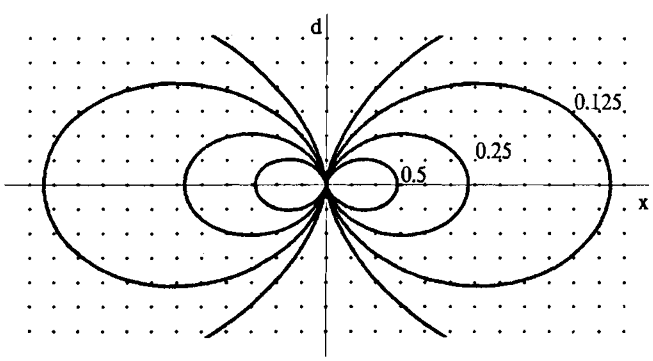
The shape of the support function used by Prazdny has a desirable fall-off for large disparity gradients (i.e. nearby candidate matches with widely different disparities), while the shape of the heat diffusion response is more elliptic (Figure 2a and 2b). These figures show a vertical cross-section through the (x,y,d) space, plotting equipotential contours of the support function. A better shape can be obtained by superimposing two heat diffusion processes. One is excitatory and has a faster diffusion in the horizontal direction, while the other is inhibitory, and has slower diffusion horizontally. This *difference of heat equations* is analogous to a difference of gaussians (DOG) function, where the superposition of two blurring functions can be used to obtain the desired response. The resulting support function approximates Prazdny's reasonably well (Figure 2c).



(a) Prazdny's function $1/c|x| \cdot exp\{-d^2/2c^2x^2\}$



(b) Heat diffusion $1/\sqrt{x^2+d^2}$ (for small α)



(c) Double heat diffusion $1/\sqrt{(\kappa_1 x)^2+d^2} - 1/\sqrt{(\kappa_2 x)^2+d^2}$

Figure 2: Shape of the support functions
The dots indicate the digital grid (i.e. possible match locations)

The other major difference between the heat equation algorithm and Prazdny's is that the latter only picks the lowest disparity gradient point in each column to give support. Prazdny's algorithm thus has the advantage that points that are *not* coherent usually do not interact. However, it is not implementable as a relaxation process.

7. Results

Simulations were made to determine the relative performance of the Prazdny and the heat equation algorithms. We made no attempt to implement the full Prazdny algorithm. In particular, instead of using edge points to determine candidate matches, only the black pixels were used. Also, the simulations were run with right-eye centered disparities only. Even so, comparative results can be obtained, by looking at the relative number of false matches made.

The algorithms were tested on a variety of sample stereograms, both transparent and opaque. The results of the three algorithms (1) Prazdny's, (2) single heat equation and (3) difference of heat equation are shown in Figure 3 for the transparent steps stereogram. Figure 4 shows the results for an opaque pyramid (wedding-cake) stereogram. These figures show the 7 disparity planes layed out horizontally (the nearest plane is to the left). The top strip in each figure shows the candidate matches (compatibilities), while the lower strips show the results for the three algorithms.

On most images, Prazdny's algorithm performed very slightly better than the difference of heat equations, and both performed significantly better than the single heat diffusion algorithm. The numerical results shown in Table 1 suggest that the difference in performance is attributable to the selection of the best match in each column to determine support (i.e. not using superposition). The shape of the support function (whether Prazdny's or difference of heat equations) matters little.

Table 1: Performance results for "transparent steps"

	Coherence	Superposition	Diffusion
Prazdny	92.9 %	88.9 %	——
Diff. of Heat	93.6 %	89.6 %	89.8 %

Notes

1. Figures are percentage correct matches out of 740.
2. Columns indicate the algorithm chosen, rows indicate the shape of the support function.
3. The diffusion equation (6 iterations) and DoH superposition results should be identical.

8. Conclusions

Recent work by Prazdny has shown that global support functions combined with a local winner-take-all selection can do well at solving the stereo correspondence problem. In general, it is impossible to implement an arbitrary global support function in a local iterative fashion. However, a function very similar to the one chosen by Prazdny can be implemented as the difference of two local heat diffusion processes. For this function, a tradeoff is available between iteration and connection complexity. Since the iterative algorithm converges quickly and has a very low connection complexity, it requires far less total computation.

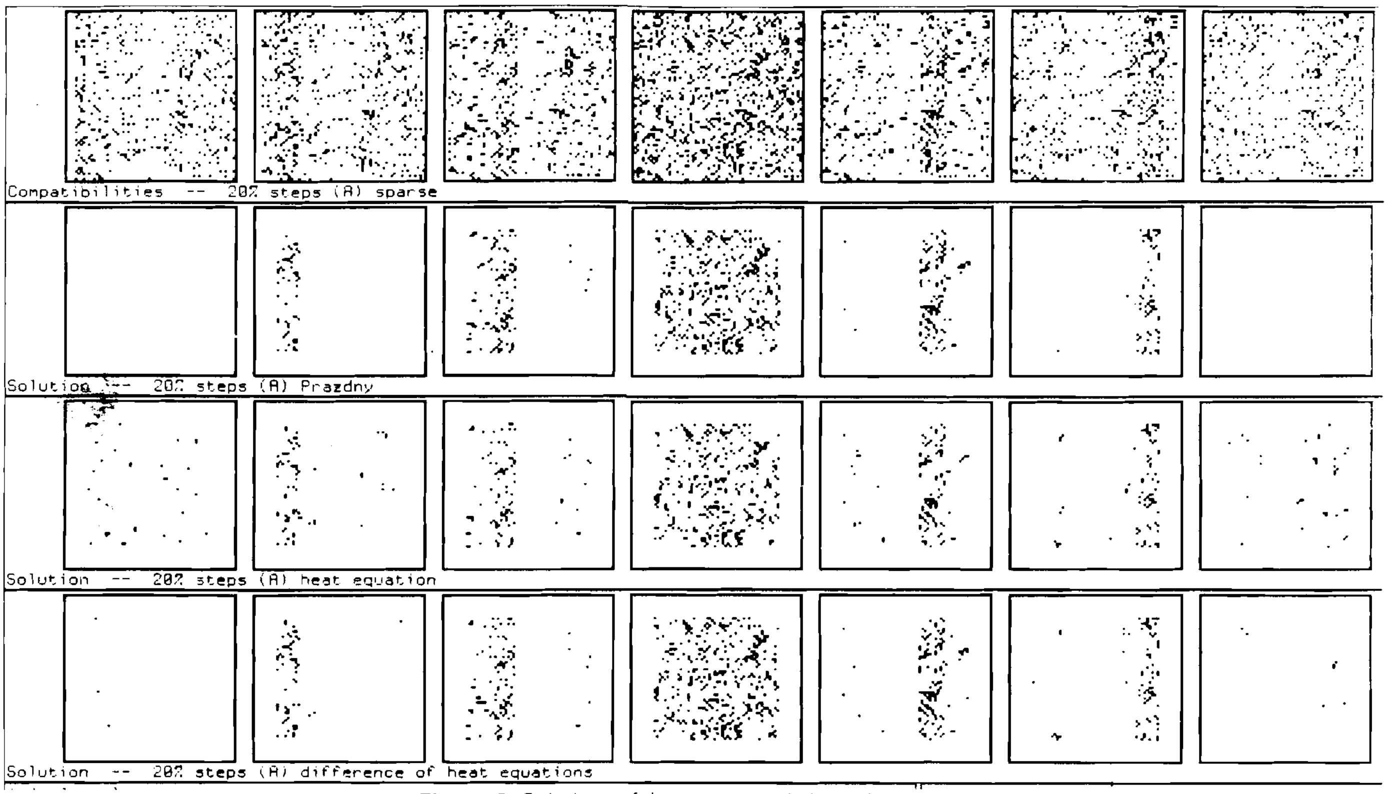


Figure 3: Solutions of the transparent steps stereogram
(a) Candidate matches (compatibilities), (b) Prazdny's algorithm,
(c) Single heat equation, (d) Double heat equation

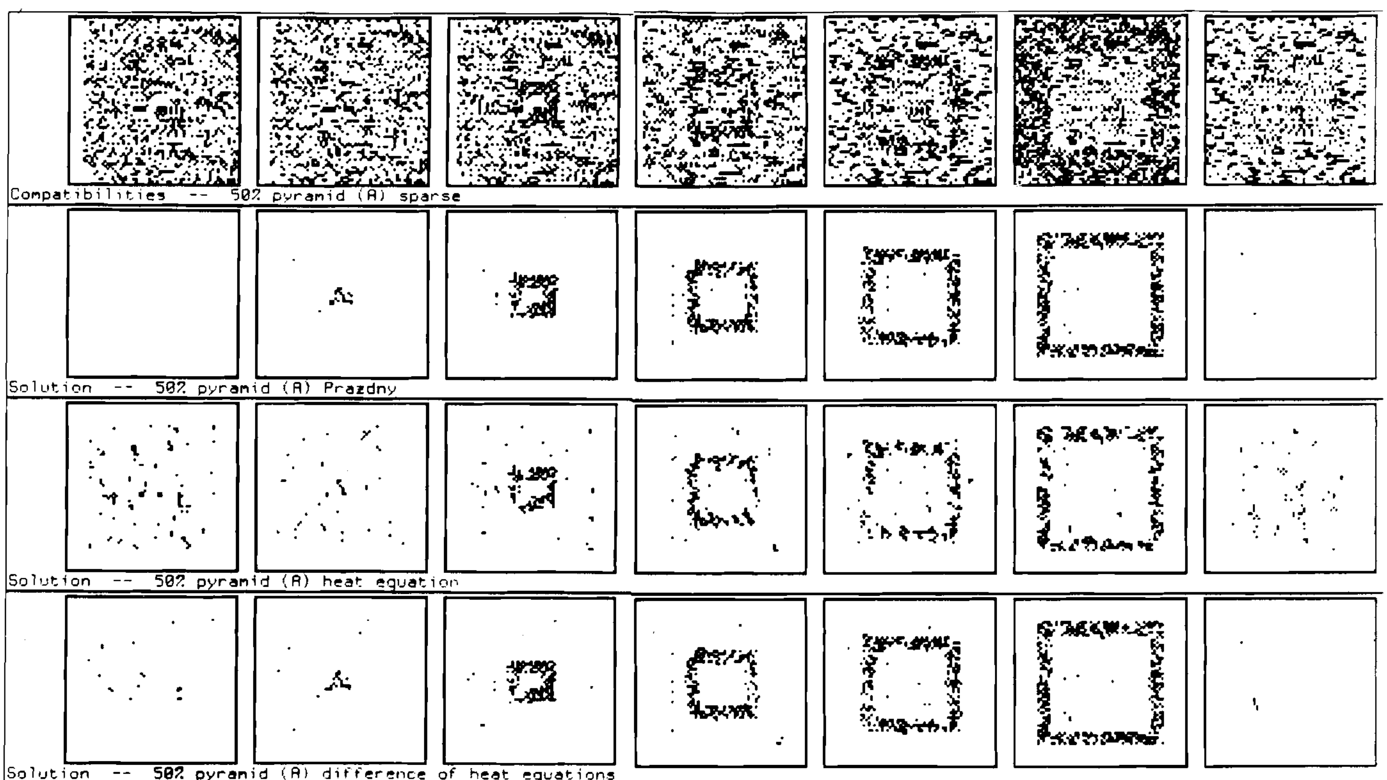


Figure 4: Solutions of the pyramid (wedding-cake) stereogram
(a) Candidate matches (compatibilities), (b) Prazdny's algorithm,
(c) Single heat equation, (d) Double heat equation

Acknowledgements

This work was supported by a grant from the System Development Foundation and a scholarship from the Allied Corporation.

References

1. K. Prazdny, "Detection of Binocular Disparities", *Biological Cybernetics*, to appear in 1985.
2. B. Julesz, "Binocular depth perception of computer generated patterns", *Bell System Technical Journal*, Vol. 39, No. 5, September 1960, pp. 1125-1162.
3. B. Julesz, *Foundations of Cyclopean Perception*, Chicago University Press, Chicago, IL, 1971.
4. G. Sperling, "Binocular Vision: A Physical and Neural Theory", *J. Am. Psychol.*, Vol. 831970, pp. 461-534.
5. P. Dev, "Segmentation Processes in Visual Perception: A Cooperative Neural Model", COINS Technical Report 74C-5, University of Massachusetts at Amherst, June 1974.
6. D. Marr and T. Poggio, "Cooperative computation of stereo disparity", *Science*, Vol. 194, October 1976, pp. 283-287.
7. R. D. Arnold, "Automated Stereo Perception", Stanford Artificial Intelligence Laboratory AIM-351, Stanford University, March 1983.
8. M. H. Kass, "Computing Stereo Correspondence", Master's thesis, Massachusetts Institute of Technology, May 1984.
9. D. Marr and T. Poggio, "A Computational Theory of Human Stereo Vision", *Proc. Royal Soc. London*, Vol. B 2041979, pp. 301-328.
10. Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 2, March 1985, pp. 139-154.
11. J. E. W. Mayhew, "A stereo correspondence algorithm using a disparity gradient limit", Personal communication
12. R. A. Hummel and B. C. Gidas, "Zero Crossings and the Heat Equation", Technical Report, Courant Insitute of Mathematical Sciences, 1984.
13. D. Terzopoulos, *Multiresolution Computation of Visible-Surface Representations*, PhD dissertation, Massachusetts Institute of Technology, January 1984.
14. W. E. L. Grimson, "An Implementation of a Computational Theory of Visual Surface Interpolation", *Computer Vision, Graphics, and Image Processing*, Vol. 221983, pp. 39-69.

A. Analytic Solution of the Heat Equation

When the only source is at the origin, u must be a function of r only, where r is the radius,

$$r = \sqrt{x^2+y^2+d^2}$$

For the stereo reconstruction problem, it may be desirable to scale d differently from x and y in its contribution to r. The derivations are similar in this extended case.

For a radially symmetric $u(r)$, the homogeneous equation is

$$\nabla^2 u - \alpha u = \frac{\partial^2 u}{\partial^2 r} + \frac{n-1}{r}\frac{\partial u}{\partial r} - \alpha u = 0$$

where n is the dimensionality of the space (here, $n=3$). This second order non-linear differential equation can be put into its standard form by using the substitution

$$u(r) = r^{-(n-1)/2} y(r)$$

which reduces to

$$y'' - [\frac{(n-1)(n-3)}{4r^2} + \alpha] y = 0$$

For three dimensions ($n=3$), the two solutions are

$$u(r) = \frac{1}{r} e^{\pm\sqrt{\alpha}\, r}$$

of which the bounded solution meets the required boundary condition $u(\infty)=0$. This solution has a discontinuity at $r=0$, so care must be taken in imposing the other boundary condition.

One way to specify the boundary condition is to have a constant temperature sphere around the origin

$$u(r) = T_0 \quad \text{for} \quad r \le r_0.$$

However, this invalidates the assumption of superposition, since heat flow is not propagated through the sphere. An alternate solution is to have the surrounding sphere be a heat source, i.e.

$$\frac{\partial u}{\partial r} = p_0 \quad \text{on} \quad r = r_0.$$

In practice, the heat source model was chosen for the finite element solution, with a point source set at the location of each candidate match.

B. Iterative algorithm implementation

The updating rule that was used was

$$L := \kappa_1 \cdot (u_{x-1,y,d} + u_{x+1,y,d} + u_{x,y-1,d} + u_{x,y+1,d}) + u_{x,y,d-1} + u_{x,y,d+1} - (2 + 4\kappa_1) \cdot u_{x,y,d};$$

$$dU := L - \alpha \cdot u_{x,y,d} + p_{x,y,d};$$

$$u_{x,y,d} := \max\langle 0, u_{x,y,d} + \Delta t \cdot dU \rangle$$

This is repeated for a second v array using κ_2, and the two results are subtracted to obtain the final support value. The match with the highest support in each disparity column is selected as the correct match.

Gauss-Seidel (i.e. asynchronous) iteration is used, with the (x,y,d) order being reversed after each pass. 6 iterations (per point) are usually sufficient for good performance. The parameters used were $\Delta t = 0.125$, $\kappa_1 = 0.75$, $\kappa_2 = 1.25$ and $\alpha = 0.25$.