Pattern Recognition Vol. 16, No. 2, pp. 223–230, 1983.
Printed in Great Britain.

0031-3203/83/020223-08 $03.00/0
Pergamon Press Ltd.

MULTI-RESOLUTION RELAXATION

K. A. NARAYANAN, DIANNE P. O'LEARY and AZRIEL ROSENFELD*
Computer Science Center and Department of Computer Science,
University of Maryland, College Park, MD 20742, U.S.A.

(*Received* 11 *November* 1981; *in revised form* 22 *April* 1982; *received for publication* 5 *May* 1982)

Abstract—Several types of iterative methods can be used to segment the pixels in an image into light and dark regions; these include "relaxation" methods of probability adjustment and steepest-descent methods of cost function minimization. Conventionally, these methods operate on the image at a single resolution. This paper investigates the possibility of using these approaches at two (or more) resolutions in order to reduce their computational cost—e.g. first obtain an approximate solution by iterating at low resolution, then refine the solution using a few iterations at high resolution.

Segmentation Relaxation Multigrid methods Pyramids

1. INTRODUCTION

Several types of iterative methods can be used to segment the pixels in an image into light and dark regions. One approach is to initially give each pixel a "light" probability, proportional to its gray level, and then use a "relaxation" algorithm to adjust these probabilities.[1] Another approach views the process as minimization of a cost function derived from the probabilities, related to both their rate of change and their entropy, and this minimization problem can be treated by steepest-descent methods.[2] Alternatively, without introducing "probabilities" at all, one can define a cost function related to both the roughness of the image and its deviation from "white" or "black", and minimize this function using steepest-descent.[3]

All of these methods conventionally operate on the image at a single resolution, with the new pixel values at each iteration dependent on the previous values of the pixel and its neighbors. It has been suggested[4,5] that there might be advantages to implementing such methods at two or more resolutions, in analogy with multigrid methods in numerical analysis.[6] For example, one might first iterate at low resolution to derive an approximate solution, then interpolate on this solution to obtain an initial estimate at high resolution, and finally iterate (a few times) at high resolution to obtain a more accurate solution.

The multiresolution approach has the possible advantage of using relatively global (= low-resolution) information in the iteration process, and in any case, it has the advantage of reduced computational cost. In this paper, we show that two-resolution methods yield results similar to those obtained at a single resolution, with substantial savings in computational cost.

Section 2 describes the experiments. The details of the algorithms used are given in the Appendix.

2. EXPERIMENTS

Our experiments made use of the two images shown in Fig. 1: an infrared image of a tank and a small portion of a Landsat scene. Two resolutions were used: 64 × 64 (the full resolution of the images) and 16 × 16.

For each of the iterative algorithms that were used, the general procedure was as follows:

(a) construct the coarse image (by 4 × 4 block averaging);

(b) perform 5–10 iterations on the coarse image;

(c) use bilinear interpolation to extend the resulting values to the fine image and combine these values with the original values in the fine image by averaging;

(d) perform 1–4 iterations on the fine image.

The results were compared with results obtained by performing up to 10 iterations directly on the fine image. Note that the coarse image has only 1/16 as many pixels as the fine image; thus, the cost of 5 to 10 iterations on the former is less than the cost of one iteration on the latter.

The first group of experiments used a steepest-descent method to minimize a two-part cost function of the form[3]

$$C = (1-\alpha)\sum_x \sum_y R_L(x,y)^2 - \alpha \sum_x \sum_y (f(x,y)-t)^2,$$

where R_L (a roughness measure) is the value of the digital Laplacian and t is the mean of the image, so that $-\Sigma\Sigma(f(x,y)-t)^2$ is low when the gray levels are all far from the mean. Figure 2 shows the results of 2, 4, 6, 8

*To whom correspondence should be addressed.

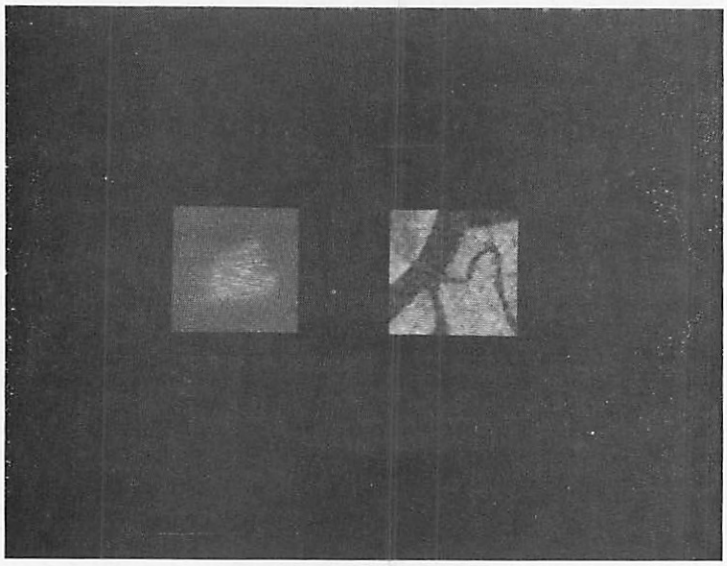

Fig. 1. Input images.

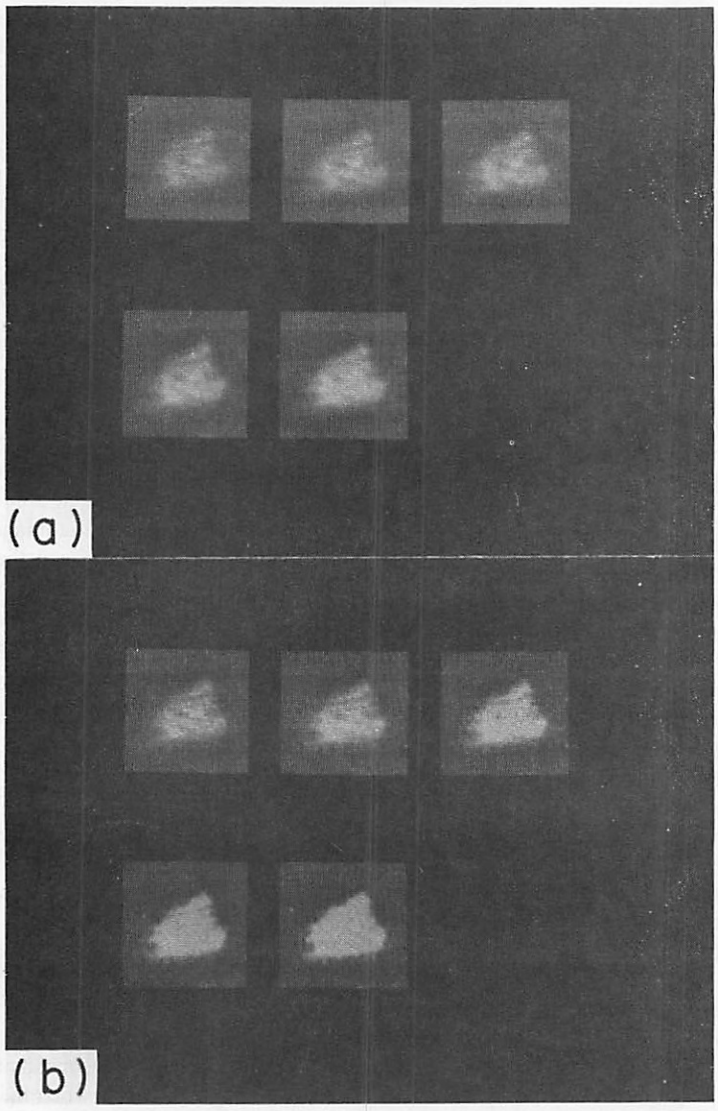

Fig. 2. Results of 2, 4, 6, 8 and 10 iterations on the fine tank image. (a) $\alpha = 0.7$; (b) $\alpha = 0.9$.

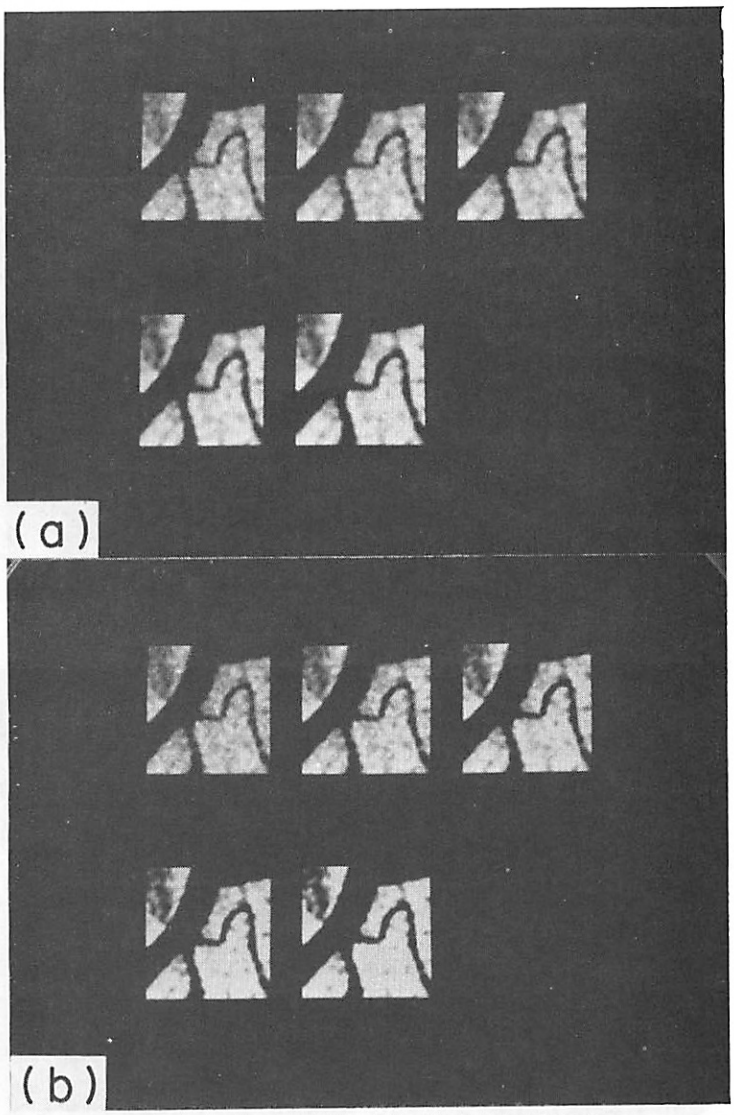

Fig. 3. Analogous to Fig. 2 for the Landsat image.

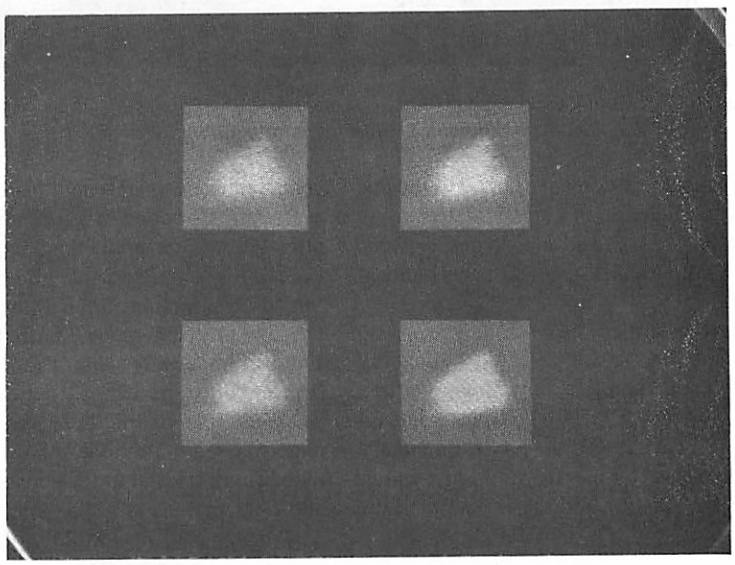

Fig. 4. Results of 2 and 4 iterations on the fine tank image using $\alpha = 0.5$, after 10 iterations on the coarse image.

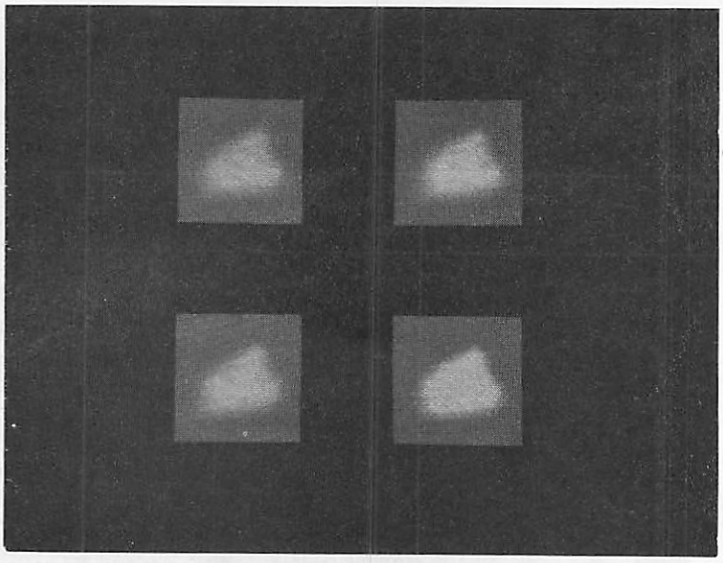

Fig. 5. Analogous to Fig. 4 for $\alpha = 0.7$.

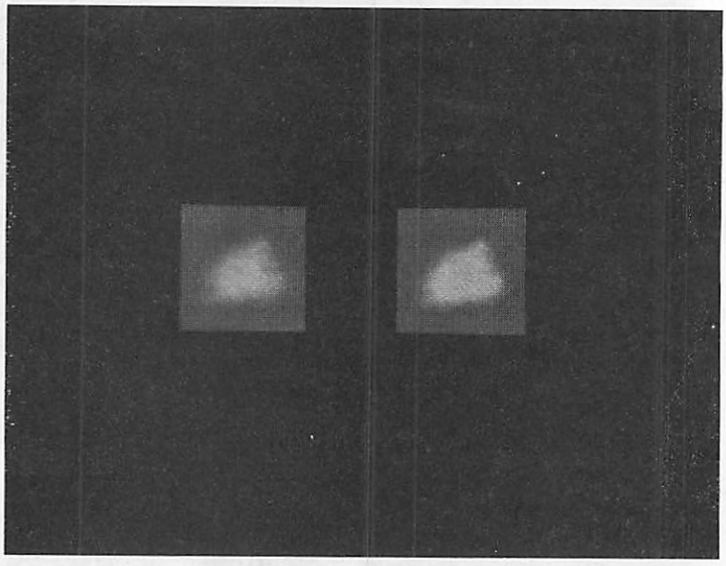

Fig. 6. Results of 4 iterations on the fine tank image using $\alpha = 0.5$, after 10 iterations on the coarse image when the latter is obtained by sampling rather than averaging.

and 10 iterations of steepest-descent applied to the fine image of the tank to minimize this cost function, using $\alpha = 0.7$ and $\alpha = 0.9$ in Fig. 2a and Fig. 2b, respectively. Figure 3a–b shows analogous results for the Landsat image.

Figures 4–8 show results of coarse/fine processing using this cost function. In all of these figures, 10 iterations were performed on the coarse image, with $\alpha = 0.7$ (left column) or $\alpha = 0.9$ (right column). Figure 4 shows results after 2 and 4 iterations on the fine image (top and bottom rows), using $\alpha = 0.5$, and Fig. 5 shows analogous results for $\alpha = 0.7$.* In these examples, the coarse image was obtained from the fine one by block averaging. For comparison, Figs. 6 and 7 show results after 4 iterations on the fine image using $\alpha = 0.5$ and $\alpha = 0.7$, respectively, when the coarse image was obtained by sampling rather than averaging; this seems to make little difference. Figure 8 shows results for the Landsat image after 4 iterations using $\alpha = 0.5$ and 0.7 (top and bottom rows), with the coarse image obtained by averaging. In all these examples, the results are more blurred than those in Figs. 2 and 3, so that some of the detail is lost, but the convergence to smooth dark and light regions is very good.

In Figs. 4–8, the initial values for the fine-image iteration were obtained by bilinearly interpolating the results of the coarse-image iterations, and then averag-

*$\alpha = 0.7$ already yields a very smooth result, suggesting that $\alpha = 0.9$ would be too high.

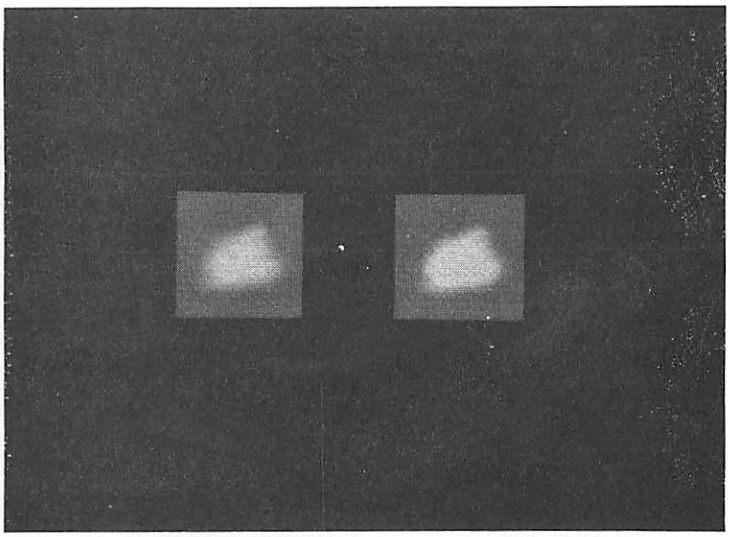

Fig. 7. Analogous to Fig. 6 for $\alpha = 0.7$.

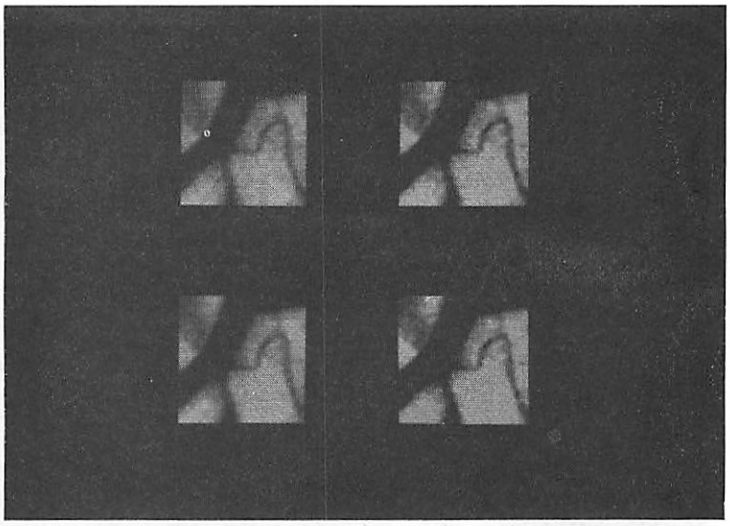

Fig. 8. Results of 4 iterations on the fine Landsat image after 10 iterations on the coarse image, using $\alpha = 0.5$ (top row) and $\alpha = 0.7$ (bottom row).

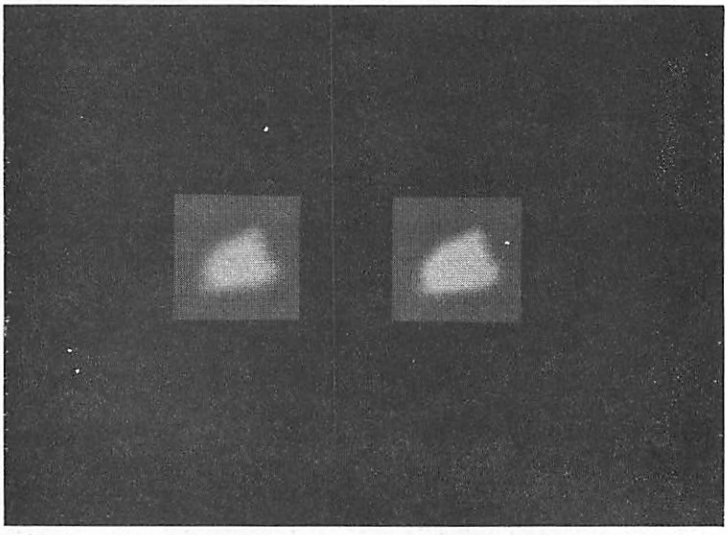

Fig. 9. Results of giving greater weight to the interpolated values than to the original fine values ($\alpha = 0.7$, 2 iterations on the fine image).

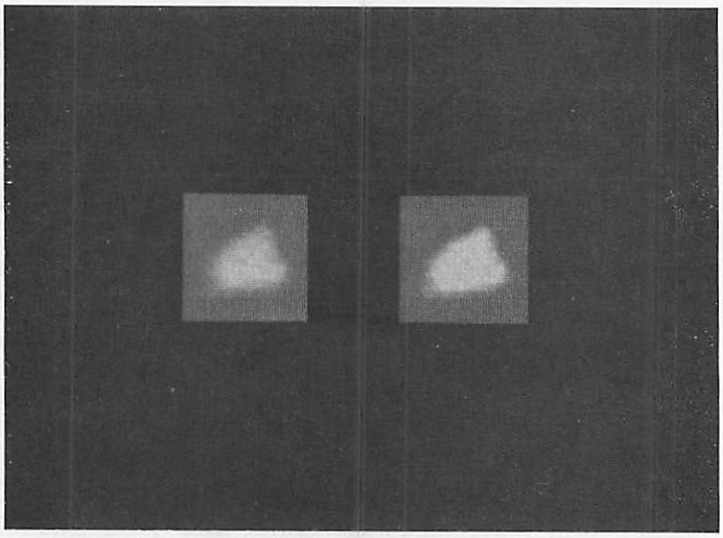

Fig. 10. Analogous to Fig. 4 (bottom row) when the fine image is obtained by nearest-neighbor rather than bilinear interpolation.

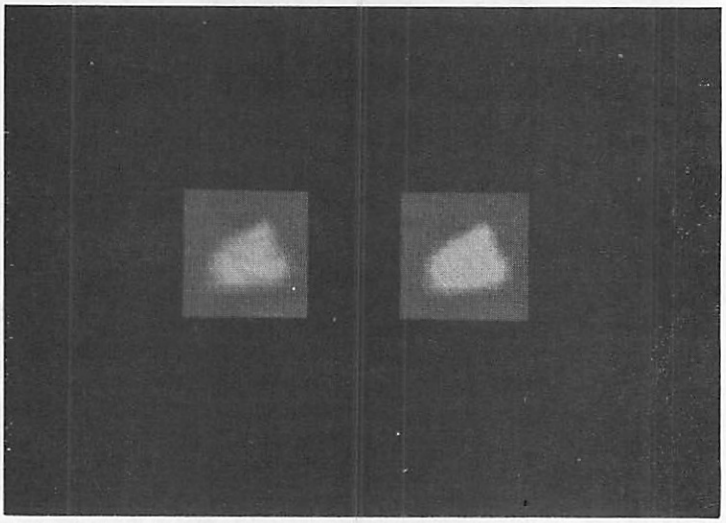

Fig. 11. Analogous to Fig. 5 (bottom row).

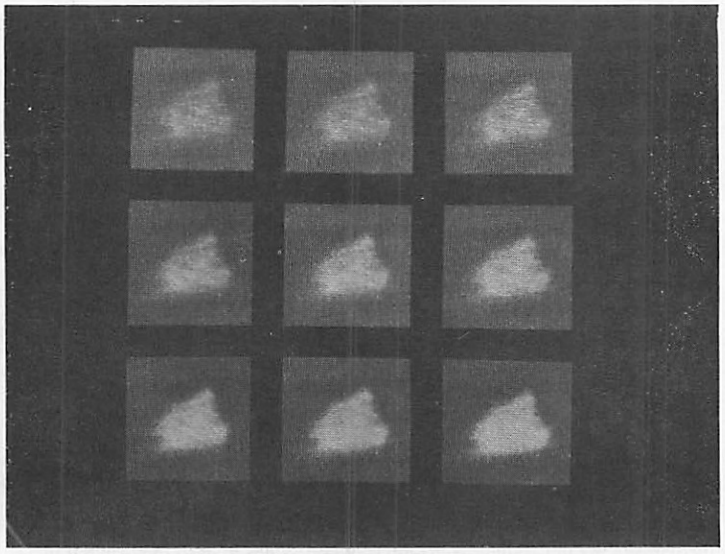

Fig. 12. Results of iterations 1–9 of probabilistic relaxation on the fine tank image.

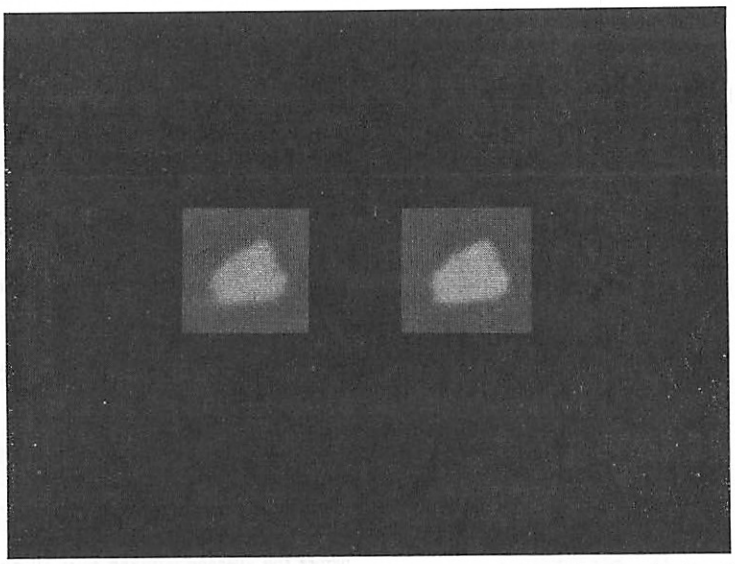

Fig. 13. Results of 1 (left) and 2 (right) iterations of probabilistic relaxation on the fine image, after 5 iterations on the coarse image.

ing these values with the original fine-image values, giving equal weight to each. Figure 9 shows the results of giving greater weight to the interpolated values (specifically, at each pixel, the interpolated value has weight 2/3 and the original fine-image value has weight 1/3). Here we used $\alpha = 0.7$ and 2 iterations. The results seem quite similar; compare the top row of Fig. 5. Figures 10 and 11 show results using nearest-neighbor, rather than bilinear, interpolation; the results are somewhat less fuzzy than those in Figs. 4–7.

The final experiment compared fine-only and coarse-fine probabilistic relaxation.[1] Figure 12 shows iterations 1–9 of relaxation (the Rosenfeld–Hummel–Zucker algorithm) applied to the tank image at full resolution. In Fig. 13, 5 iterations of this relaxation algorithm were applied to the coarse image (obtained by block averaging) and then 1 or 2 iterations (left and right columns) of the same algorithm were applied to the fine image, after initializing the probabilities by bilinear interpolation. Again, there is some loss of detail, but the convergence is very good at very low computational cost (compare Fig. 12 with iterations 2 and 3 in Fig. 9).

3. CONCLUDING REMARKS

Two-resolution methods of relaxation and cost function minimization can be used to segment an image into light and dark regions at much lower computational cost than single-resolution methods, though with loss of fine detail. Thus, the two-(or multi-) resolution approach may be preferable to conventional methods, especially in situations where multiresolution image representations ("pyramids") are already being used for other purposes.

A possibility for further reducing computational cost in a two-resolution scheme would be to apply the process to the fine image only in border regions, i.e. in the vicinity of coarse image points at which the roughness measure remains high.

Acknowledgements—The support of the National Science Foundation under Grant MCS-79-23422 is gratefully acknowledged, as is the help of Janet Salzman in preparing this paper.

REFERENCES

1. R. Smith and A. Rosenfeld, Thresholding using relaxation, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-3,** 598–606 (1981).
2. O. Faugeras and M. Berthod, Scene labeling: an optimization approach, *Pattern Recognition* **12,** 339–347 (1980).
3. K. A. Narayanan, D. P. O'Leary and A. Rosenfeld, Image smoothing and segmentation by cost minimization, *IEEE Trans. Syst. Man Cybernet.* **SMC-12,** 91–96 (1982).
4. A. Danker and A. Rosenfeld, Blob detection by relaxation, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-3,** 79–92 (1981).
5. D. H. Ballard, Parameter networks: towards a theory of low-level vision, Computer Science Department report TR-75, University of Rochester, New York (1981).
6. A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Maths Comput.* **31,** 333–390 (1977).

APPENDIX. ALGORITHMS

A.1. *Cost function minimization*

In the cost function

$$(1-\alpha)\sum_x\sum_y R_L(x,y)^2 - \alpha\sum_x\sum_y (f(x,y)-t)^2,$$

the roughness measure R_L used was the value of the digital Laplacian

$$L(f) \equiv 4f(x,y) - [f(x-1,y) + f(x,y-1) + f(x+1,y) + f(x,y+1)]$$

and t was the mean of the image f. The steepest-descent method minimizes this cost function by constructing a sequence of images $f \equiv f^{(0)}, f^{(1)}, f^{(2)}, \ldots$, constructing $f^{(k+1)}$ by adjusting $f^{(k)}$ at each point in a direction that reduces

$$C^{(k)} \equiv (1-\alpha) \sum_x \sum_y L(f^{(k)})^2 + \alpha \sum_x \sum_y (f^{(k)}(x,y) - t)^2$$

subject to the restriction that the values of $f^{(k)}$ remain within the allowed gray level range. Specifically, we have

$$f^{(k+1)}(x,y) = f^{(k)}(x,y) - \lambda^{(k)} \partial C^{(k)}(x,y)/\partial f^{(k)}$$

It can be verified that for the Laplacian roughness measure we have

$$\begin{aligned}\frac{1}{2}\frac{\partial C^{(k)}}{\partial f^{(k)}(x,y)} = {} & 20f(x,y) - 8[f(x-1,y) + f(x,y-1) \\ & + f(x+1,y) + f(x,\ y+1)] \\ & + 2[f(x-1,\ y-1) + f(x-1,\ y+1) \\ & + f(x+1, y-1) + f(x+1, y+1)] \\ & + [f(x-2,\ y) + f(x,\ y-2) \\ & + f(x+2,\ y) + f(x,\ y+2)] \\ & - \alpha(f^{(k)}(x,\ y) - t)\end{aligned}$$

A.2. *Probabilistic relaxation*

The initial gray level z_i at the i-th pixel is mapped into initial "light" and "dark" probabilities as follows: let d, l be the darkest and lightest gray levels; then $p_{i,l}^{(0)} \equiv p_{i,\text{dark}}^{(0)} = (l - z_i)(l - d)$; $p_{i2}^{(0)} \equiv p_{i,\text{light}}^{(0)} = (z_i - d)(l - d)$. These probabilities are then iteratively adjusted using the formula

$$p_{ij}^{(k+1)} = p_{ij}^{(k)}(1 + q_{ij}^{(k)}) / \sum_{j=1}^{2} p_{ij}^{(k)}(1 + q_{ij}^{(k)})$$

where $q_{ij}^{(k)}$ is the average, over all neighbors u of the i-th pixel, of

$$\sum_{v=1}^{2} C(i,j;u,v) p_{uv}^{(k)}.$$

Here $C(i,j;u,v)$ was taken to be

$$\frac{\text{Av}(p_{ij}^{(0)} p_{uv}^{(0)})}{\text{Av}(p_{ij}^{(0)})\,\text{Av}(p_{uv}^{(0)})},$$

where the average is taken over all pixel pairs in the image having the same neighbor relationship as u has to i.

About the Author—K. A. NARAYANAN was born on 4 October 1949 in India. He received his BE in Electronics in 1970 from Bangalore University, India, and his ME in Electrical Communication Engineering in 1972 from the Indian Institute of Science, Bangalore, India. Since 1972 he has been working as an engineer at ISRO Satellite Centre, Bangalore, India. Since July 1980 he has also been engaged in research at the Computer Vision Laboratory, University of Maryland. His research interests include digital signal processing and digital image processing.

About the Author—DIANNE P. O'LEARY received the BS degree in Mathematics from Purdue University in 1972 and the PhD degree in Computer Science from Stanford University in 1976. She was an assistant professor of mathematics at the University of Michigan from 1975 to 1978 and has been an assistant professor of computer science at the University of Maryland since then. Since 1978 she has also been a consultant to the United States National Bureau of Standards. Her research interests include numerical analysis and its applications, with emphasis on computational linear algebra and optimization.

About the Author—AZRIEL ROSENFELD received the PhD in Mathematics from Columbia University in 1957. After ten years in the defense electronics industry, in 1964 he joined the University of Maryland, where he is Research Professor of Computer Science. He edits the journal *Computer Graphics and Image Processing*, and is president of the consulting firm ImTech, Inc. He has published 13 books and over 250 papers, most of them dealing with the computer analysis of pictorial information. He is currently President of the International Association for Pattern Recognition.