Pattern Recognition Letters 12 (1991) 619–626
North-Holland

October 1991

Textural analysis of range images

Sunil Arya, Daniel DeMenthon, Peter Meer and Larry S. Davis

Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD 20742-3411, USA

Received 28 November 1990
Revised 9 April 1991

Abstract

Arya, S., D. DeMenthon, P. Meer and L.S. Davis, Textural analysis of range images, Pattern Recognition Letters 12 (1991) 619–626.

This paper addresses the problem of texture discrimination in range images. The range data is transformed to a common coordinate system, and then resampled to generate a regular grid of points in order to eliminate the effect of different sampling rates. Eight statistical textural features based on co-occurrence matrices are computed on the resampled data, and are used to discriminate between two classes of natural surfaces consisting of pebbles of different sizes lying on a plane. Seven of the eight textural features are found to be useful in discriminating between these two classes. The experiments also confirm the importance of resampling the data before computing the textural features.

Keywords. Texture analysis.

1. Introduction

This paper addresses the problem of texture discrimination in range data. Our long term goal is to be able to segment a range image into regions of uniform range texture (such as shrubs versus grass) for purposes of visual cross country navigation.

The problem of identification and description of texture in grey level images has been extensively studied (e.g. see survey by Haralick, 1979). It has also been pointed out that texture analysis techniques developed for grey level images are applicable, in principle, to three-dimensional range data (Morgenthaler et al., 1982). However, most of the methods developed for describing texture in grey level images do not take into account the fact that the image characteristics of textured surfaces depend upon:

(1) the relative orientation of the textured surface and the image plane, and

(2) the distance between the textured surface and the image plane.

Much work has also been done on the problem of using texture in grey level images as a cue for computing the orientation and depth of the textured surface. Two different kinds of general techniques have been used for this purpose. The first deduces orientation using texture gradients such as the systematic variation in the density of texture elements in the perspective projection of a slanted textured surface (Rosenfeld, 1975; Aloimonos and Swain, 1985; Kanatani and Chou, 1986). The second uses purely local properties such as the distor-

The support of the Defence Advanced Research Projects Agency (ARPA Order No. 6350) and the U.S. Army Engineer Topographic Laboratories under Contract DACA76-88-C-0008 is gratefully acknowledged.

0167-8655/91/$03.50 © 1991 — Elsevier Science Publishers B.V. All rights reserved

tion in the distribution of edge directions caused by projective foreshortening (Witkin, 1981; Davis et al., 1983; Kanatani, 1984). However, when range information is available the problem of determining the orientation and depth of the textured surface is simplified. It is possible to use this information to 'correct' image patches for range and orientation and, after suitable transformations, compute their textural properties. The problems that must be addressed to perform these corrections are:

(1) transforming the data to a common coordinate system,

(2) resampling of the data in order to generate a regular grid of points, and

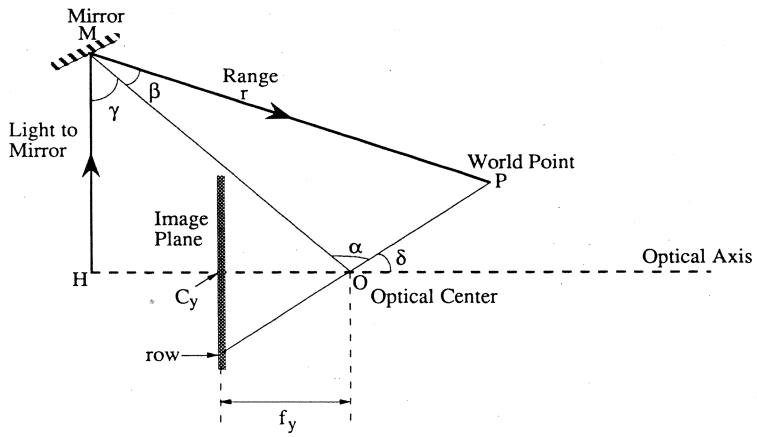
(3) fitting surfaces to textured patches.

We tried to distinguish between two classes of natural surfaces for our experiments. The first class consisted of small pebbles (size about 1 cm) lying on a plane, and the second class consisted of somewhat larger pebbles (size about 1.5 cm). For each of these two classes, eight range images were taken, the relative position of the range scanner and the plane being different for each image. The CVL Light-Stripe Range Scanner, described in DeMenthon et al. (1987), was used to produce the images (Section 2). Since the range scanner samples different parts of a surface at different sampling intervals, the range image contains information about points which are not spaced at regular intervals.

A plane is fitted to the points represented by the range image, using the robust least median of squares method (Section 3), and the residual from the plane is computed at each point. This yields the *non-uniformly sampled range image* which consists of the residuals from the plane for different points on the plane (Section 4.1). The points on the plane are spaced at irregular intervals, usually at a distance of 1–2 mm from one another. The next step is to compute a *uniformly sampled range image* which consists of the residuals from the plane for points on the plane at regular intervals (Section 4.2). Finally we compute textural features from co-occurrence matrices for the uniformly sampled range image, and use them to discriminate between the two classes of natural surfaces (Section 5). The experimental results are presented in Section 6, and the concluding remarks are given in Section 7.

2. Creating range images

The production of range images with the CVL Light-Stripe Range Scanner has been described in DeMenthon et al. (1987). Here we briefly outline the method. The ranging instrument comprises a light-emitting slit, a step-motor controlled mirror and a CCD camera compact enough to be mounted on the tool plate of a robot arm. The stripe of light produced by a laser source is incident on the mir-



$$r = \frac{OM \sin \alpha}{\sin(\alpha + \beta)} \qquad \alpha = \Pi/2 + \gamma + \delta \qquad \delta = \arctan\left(\frac{row - C_y}{f_y}\right)$$

Figure 1. Calculating the range.

Table 1
Sampling intervals for range images of smaller pebbles

	row-interval (mm)	column-interval (mm)	angle (degree)
image 1	0.87–1.95	0.65–1.95	22–52
image 2	1.04–1.52	1.07–1.88	21–40
image 3	1.21–1.70	1.25–1.95	27–41
image 4	1.12–1.61	1.14–1.88	26–41
image 5	1.11–1.83	0.99–1.89	29–47
image 6	0.96–1.34	0.97–1.81	8–33
image 7	1.38–1.64	1.66–2.10	24–33
image 8	1.19–1.61	1.49–2.39	4–31

Table 2
Sampling intervals for range images of larger pebbles

	row interval (mm)	column interval (mm)	angle (degree)
image 1	1.12–1.83	1.10–2.06	25–45
image 2	1.08–1.81	1.11–2.16	20–44
image 3	1.25–1.61	1.53–2.18	17–33
image 4	1.07–1.92	1.05–2.16	23–47
image 5	1.32–1.53	1.69–2.14	15–26
image 6	0.99–1.97	0.88–2.13	22–49
image 7	1.09–1.70	1.23–2.23	14–39
image 8	1.06–2.03	0.97–2.15	26–50

ror, which is rotated to project the light-sheet at varying angles. For each position of the mirror, the ranges of the world points are placed in the same row of the range image. The light-sheet intersects the world scene in a planar stripe whose image is detected in the camera image plane. Figure 1 shows how the range of a world point P with respect to the mirror is computed given the position of the mirror and the projection of the world point P on the image plane. The 512×512 range image produced by this system contains the ranges of the world points with respect to the mirror axis, and this is resampled to yield a 256×128 image. That portion of this image which contains the world points for the textured surface under consideration is referred to as the *raw range image*.

Tables 1 and 2 give the approximate sampling rates along the rows and columns for each of the eight range images corresponding to the small pebbles (Figure 2a), and for each of the eight range images corresponding to the large pebbles (Figure 2b), respectively. The *angle* column in these tables gives the angle that the light stripe makes with the textured surface.

3. Robust lead median of squares method

Least squares regression, while commonly employed in computer vision, is highly unreliable when some of the data deviates severely from the linear relation followed by the majority of the

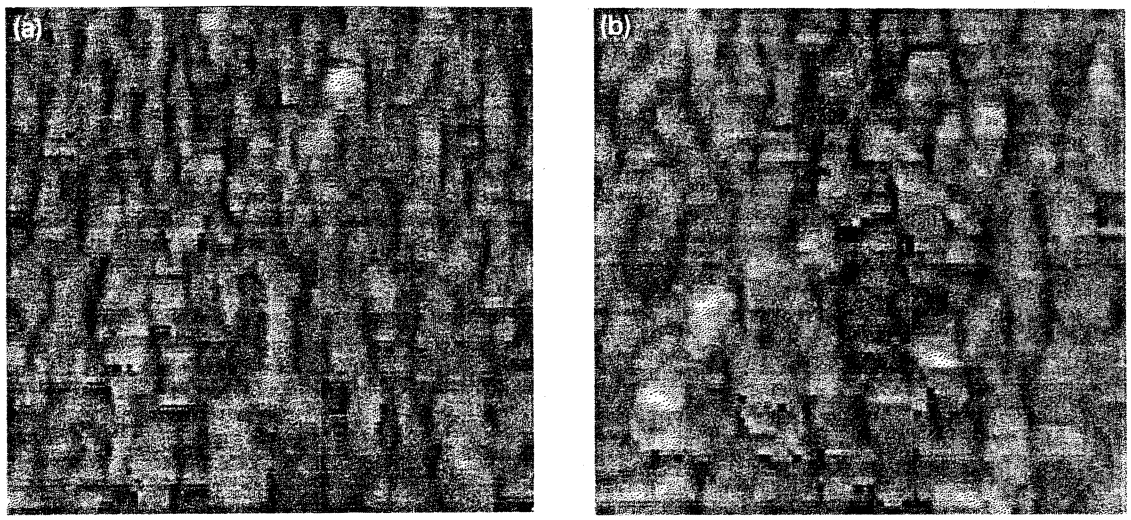


Figure 2. Examples of non-uniformly sampled range images. (a) Small pebbles. (b) Large pebbles.

data. Hence the need for robust methods which can work well even in the presence of *outliers* (data points that yield significant deviations). These outliers need not necessarily be error points; for example, when two surfaces meet in a window then we can regard the surface generating the majority of points in the window as the signal and the remaining points as outliers. The breakdown point of a regression method is the smallest fraction of contamination of the data which yields arbitrarily incorrect estimates. The Least Median of Squares (LMedS) method has the additional advantage over other robust methods of having a high breakdown point of 50%. Rousseeuw (1984) proposed the LMedS method in which the parameters are estimated by solving the minimization problem

$$\min_i \operatorname{med} r_i^2. \tag{1}$$

Here the residual r_i is the difference for the ith point between the data actually observed and the estimated fit on the basis of the model. This does not have a closed-form solution; it is solved by generating possible solutions and searching among them to find the best one (Meer et al., 1990).

We now describe the process of fitting a plane using this method to a set of n data points denoted by indices $i_1, \ldots, i_n$. Let the plane be denoted by

$$z = \beta_0 + \beta_1 x + \beta_2 y. \tag{2}$$

For any set of three points denoted by indices i, j, k, β_1 and β_2 are computed directly using the following planar fit equations:

$$\beta_1 = \frac{(z_i - z_j)(y_j - y_k) - (z_j - z_k)(y_i - y_j)}{(x_i - x_j)(y_j - y_k) - (x_j - x_k)(y_i - y_j)}, \tag{3}$$

$$\beta_2 = \frac{(z_i - z_j)(x_j - x_k) - (z_j - z_k)(x_i - x_j)}{(y_i - y_j)(x_j - x_k) - (y_j - y_k)(x_i - x_j)}. \tag{4}$$

β_0 is then computed so as to solve the following minimization problem:

$$\min_i \operatorname{med} r_i^2 \quad \text{given } \beta_1 \text{ and } \beta_2. \tag{5}$$

This procedure may then be repeated for every other set of three points, and the values of $\beta_0, \beta_1, \beta_2$ chosen are those that correspond to the smallest value for (5).

Steele and Steiger (1986) proposed the following method to solve (5). Define

$$s_i = z_i - \beta_1 x_i - \beta_2 y_i,$$

and reorder the set of indices such that

$$s_1 \leqslant s_2 \leqslant \cdots \leqslant s_n.$$

Define M_i and D_i as follows:

$$M_i = \frac{(s_{i+\lfloor n/2 \rfloor} + s_i)}{2} \quad \text{for } 1 \leqslant i \leqslant \left\lceil \frac{n}{2} \right\rceil, \tag{6}$$

$$D_i = \frac{(s_{i+\lfloor n/2 \rfloor} - s_i)}{2} \quad \text{for } 1 \leqslant i \leqslant \left\lceil \frac{n}{2} \right\rceil. \tag{7}$$

The functions $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are the floor and ceiling functions respectively. It can be seen that the following relation holds for any k:

$$D_k = M_k - s_k = s_{k+\lfloor n/2 \rfloor} - M_k. \tag{8}$$

The following two ordering relations also hold for any k:

$$|M_k - s_i| \begin{cases} \leqslant D_k & \text{if } k < i < k + \lfloor n/2 \rfloor, \\ \geqslant D_k & \text{if } 1 \leqslant i < k \text{ or} \\ & \quad k + \lfloor n/2 \rfloor < i \leqslant n. \end{cases} \tag{9}$$

From the above two ordering relations it follows that for any k:

$$D_k = \operatorname{med} |M_k - s_i|. \tag{10}$$

Next we write an expression for the median of the residuals when M_k is chosen as the intercept, that is, when $\beta_0 = M_k$:

$$\begin{aligned} \operatorname{med} |r_i| &= \operatorname{med} |z_i - b_1 x_i - b_2 y_i - M_k| \\ &= \operatorname{med} |M_k - s_i|. \end{aligned} \tag{11}$$

Combining (11) with (10), we get that when $\beta_0 = M_k$,

$$\operatorname{med} |r_i| = D_k. \tag{12}$$

Assuming that the optimal value of the intercept β_0 is one of the M_i's, then clearly, to minimize (5), we must choose $\beta_0 = M_l$ where l satisfies the following equation:

$$D_l = \min_i D_i. \tag{13}$$

It is beyond the scope of the paper to show that the optimal value of β_0 must be one of the M_i's; for a proof of this see Rousseeuw (1987, p. 166).

Carrying out the above procedure for every set of three points makes it computationally very ex-

pensive since there are $C(n, 3)$ such triplets and for each of them n values have to be sorted yielding a time complexity of $O(n^4 \log n)$. However, if we are willing to tolerate a small probability of error E, then it is not necessary to consider every set of three points. If only m sets of three points are considered, then the probability that every one of these sets contains an outlier is

$$E = [1-(1-f)^3]^m \tag{14}$$

where f is the fraction of points that are outliers. For our experiments we choose $m = 36$ which yields a probability of error of less than a percent even when the fraction of outliers is as high as 49.9%.

4. Processing the range image

The advantage of computing the textural features on the uniformly sampled range image in preference to the non-uniformly sampled range image is that it eliminates the effect of different sampling rates. The creation of the uniformly sampled range image from the raw range image involves two steps. First we transform the range data to a common coordinate system yielding the non-uniformly sampled range image. Next we resample the range data to generate a regular grid of points yielding the uniformly sampled range image.

4.1. Creating the non-uniformly sampled range image

A plane P is fitted to the entire set of points lying on the textured surface, using the robust least median of squares method, described in the previous section. The set of points is next transformed to a reference frame which has x-axis and y-axis on the plane P, and z-axis normal to the plane P. The non-uniformly sampled range image $NU(i, j)$ then consists of the residuals (the z-values) from the plane P, for points lying at irregular intervals on the plane P.

4.2. Creating the uniformly sampled range image

A uniformly sampled range image is created which consists of the z-values (residuals) at a rectangular grid $N_x \times N_y$ of points lying on the least median of squares plane P. The grid interval, δ, was chosen to be the same for both the x and the y directions. We chose δ to be 2.5 mm in order to be greater than the sampling interval in the range image (1–2 mm), and it was kept the same for all the 16 uniformly sampled range images.

The residual at each point of the grid is computed using a robust interpolation technique. For each grid point we find 25 of its nearest neighbors on the non-uniformly sampled range image and fit a plane to them using the robust least median of squares method described earlier. The residual at the grid points is found by interpolation from the fitted plane.

5. Computation of textural features

For visual images the computation of textural features using spatial co-occurrence matrices has been described in Haralick (1979). Our method for computing texture features on uniformly sampled range images is analogous. The uniformly sampled range image consists of the residuals from the least median of squares plane P on a rectangular grid with N_x positions along the x-direction and N_y positions along the y-direction. The residuals are quantized into N_g levels where each level denotes an interval of size 1 mm. The co-occurrence $p(i, j)$ of quantized residuals i and j for an image I is defined as the number of pairs of grid points having quantized residuals i and j, respectively, and which bear a fixed relationship to one another on the grid. Four different co-occurrence matrices that correspond to four different relationships on the grid are used in the experiments. Each of these corresponds to a distance of one grid unit, but to different angles of 0° (horizontal nearest neighbor), 90° (vertical nearest neighbor), 45° and 135° (nearest neighbors along the two diagonals, respectively). Formally, these four different co-occurrence matrices are defined in Haralick et al. (1973):

$$\begin{aligned} p(i, j, 0^\circ) &= \#\{((k, l), (m, n)) \mid k-m=0,\ |l-n|=1, \\ &\qquad I(k, l)=i,\ I(m, n)=j\}, \end{aligned}$$

Table 3
Textural features of uniformly sampled range images of smaller pebbles

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
image 1	0.05	2.60	0.33	1.94	0.55	3.34	1.13	1.36
image 2	0.04	3.56	0.16	2.12	0.49	3.48	1.48	1.47
image 3	0.04	3.28	0.37	2.61	0.51	3.62	1.40	1.46
image 4	0.05	3.20	0.18	1.94	0.50	3.39	1.25	1.42
image 5	0.05	2.90	0.25	1.93	0.52	3.39	1.20	1.40
image 6	0.04	3.36	0.32	2.46	0.49	3.62	1.36	1.46
image 7	0.04	5.75	0.11	3.24	0.47	3.70	2.82	1.62
image 8	0.04	3.59	0.29	2.52	0.49	3.64	1.50	1.49

$$p(i,j,45°) = \#\{((k,l),(m,n)) \mid (k-m=1,\ l-n=-1) \text{ or } (k-m=-1,\ l-n=1),\ I(k,l)=i,\ I(m,n)=j\},$$

$$p(i,j,90°) = \#\{((k,l),(m,n)) \mid |k-m|=1,\ l-n=0,\ I(k,l)=i,\ I(m,n)=j\},$$

$$p(i,j,135°) = \#\{((k,l),(m,n)) \mid (k-m=1,\ l-n=1) \text{ or } (k-m=-1,\ l-n=-1),\ I(k,l)=i,\ I(m,n)=j\}$$

where $((k,l),\ (m,n)) \in (N_x \times N_y) \times (N_x \times N_y)$.

The co-occurrence matrix is normalized by dividing each entry by the sum of all the entries in the matrix, to yield the normalized co-occurrence matrix. Eight textural features, all of which were taken from Appendix 1 of Haralick et al. (1973), are computed for each of the four normalized co-occurrence matrices. The textural features used are Angular Second Moment (f_1), Contrast (f_2), Correlation (f_3), Sum of Squares (f_4), Inverse Difference Moment (f_5), Entropy (f_6), Difference Variance (f_7), and Difference Entropy (f_8). Each of these textural features is averaged over its four values computed for the four normalized co-occurrence matrices. The averaging is done so as to ensure that the textural features have the same values for two images that are identical except that one is rotated with respect to the other.

6. Experimental results

Tables 3 and 4 show the results of computing the eight textural features on the eight uniformly sampled range images corresponding to the class of small and large pebbles, respectively. It can be seen from these two tables that, with the exception of the seventh image of the small pebbles, the follow-

Table 4
Textural features of uniformly sampled range images of larger pebbles

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
image 1	0.03	5.72	0.14	3.31	0.42	3.90	2.14	1.65
image 2	0.03	5.72	0.21	3.63	0.42	3.98	2.21	1.65
image 3	0.02	6.08	0.33	4.52	0.42	4.14	2.40	1.68
image 4	0.03	5.15	0.17	3.10	0.45	3.83	2.17	1.62
image 5	0.02	7.83	0.27	5.36	0.38	4.34	2.98	1.79
image 6	0.02	6.77	0.20	4.23	0.41	4.07	2.67	1.71
image 7	0.02	6.65	0.26	4.50	0.41	4.17	2.65	1.72
image 8	0.03	5.99	0.14	3.48	0.43	3.93	2.39	1.67

Table 5
Textural features of non-uniformly sampled range images of smaller pebbles

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
image 1	0.06	1.11	0.77	2.46	0.71	3.16	0.67	1.01
image 2	0.04	1.75	0.71	3.06	0.61	3.52	0.89	1.19
image 3	0.03	2.38	0.70	4.03	0.57	3.80	1.16	1.32
image 4	0.04	1.66	0.74	3.20	0.62	3.54	0.83	1.18
image 5	0.05	1.47	0.74	2.85	0.65	3.41	0.78	1.14
image 6	0.04	2.18	0.72	3.87	0.62	3.66	1.25	1.23
image 7	0.03	3.00	0.69	4.79	0.53	3.97	1.43	1.41
image 8	0.03	2.83	0.70	4.80	0.54	4.00	1.34	1.39

ing features have no overlap for the two tables — Angular Second Moment (f_1), Contrast (f_2), Sum of Squares (f_4), Inverse Difference Moment (f_5), Entropy (f_6), Difference Variance (f_7), and Difference Entropy (f_8). These features therefore have the potential of being used to discriminate between the two classes. In order to examine the importance of computing the uniformly sampled range images, we also computed the same eight textural features on the non-uniformly sampled range images. (The computation of co-occurrence matrices in the case of non-uniformly sampled range images is analogous to the computation for uniformly sampled range images; instead of nearest neighbors on a regular grid, we now have nearest neighbors between resolution cells.) Tables 5 and 6 show the results of computing these textural features on the non-uniformly sampled range images corresponding to the class of small and large pebbles, respectively. It can be seen from these two tables that all the features have a large overlap. This empirically demonstrates the importance of computing textural features using the uniformly sampled range images instead of the non-uniformly sampled range images.

7. Concluding remarks

We have described a method for classifying textures in range images using statistical textural features based on co-occurrence matrices. Before being able to compute these textural features, we have to transform the range data to a common coordinate system and resample the data in order to generate a regular grid of points. In our initial experiments seven of the features have proven to be of use. We intend to carry out these experiments with more classes of textures. The long term goal is to be able to segment a range image into regions of uniform range texture for purposes of visual cross country navigation.

Table 6
Textural features of non-uniformly sampled range images of larger pebbles

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
image 1	0.03	1.85	0.80	4.55	0.62	3.75	0.99	1.21
image 2	0.03	2.17	0.78	4.95	0.60	3.84	1.18	1.25
image 3	0.02	3.41	0.75	6.86	0.52	4.24	1.67	1.45
image 4	0.03	2.00	0.80	4.94	0.62	3.78	1.12	1.21
image 5	0.02	4.27	0.75	8.53	0.47	4.48	1.97	1.55
image 6	0.03	1.83	0.83	5.39	0.64	3.79	1.04	1.18
image 7	0.02	2.64	0.82	7.37	0.57	4.16	1.38	1.34
image 8	0.04	1.77	0.81	4.61	0.63	3.72	0.97	1.18

References

Aloimonos, J. and M. Swain (1985). Shape from texture. *Proc. 9th Internat. Joint Conf. on Artificial Intelligence*, August 1985, 926–931.

Davis, L.S., L. Janos and S. Dunn (1983). Efficient recovery of shape from texture. *IEEE Trans. Pattern Anal. Machine Intell.* 5(5), 485–492.

DeMenthon, D., T. Siddalingaiah and L.S. Davis (1987). Production of dense range images with the CVL light-stripe range scanner. *CAR-TR-337*, Computer Vision Lab., University of Maryland, College Park, MD.

Haralick, R.M. (1979). Statistical and structural approaches to texture. *Proc. IEEE* 67(5), 786–804.

Haralick, R.M., K. Shanmugam and I. Dinstein (1973). Textural features for image classification. *IEEE Trans. Syst. Man Cybernet.* 3(6), 610–622.

Kanatani, K. (1984). Detection of surface orientation and motion from texture by a stereological technique. *Artificial Intelligence* 23, 213–237.

Kanatani, K. and T. Chou (1986). Shape from texture: General principles. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 1986, 578–583.

Meer, P., D. Mintz and A. Rosenfeld (1990). Least median of squares based robust analysis of image structure. *CAR-TR-490*, Computer Vision Lab., University of Maryland, College Park, MD.

Morgenthaler, D.G., C.-Y. Wang and A. Rosenfeld (1982). Two remarks on multidimensional texture analysis. *Pattern Recognition Letters* 1, 103–105.

Rosenfeld, A. (1975). A note on automatic detection of texture gradients. *IEEE Trans. Computers* 24, 988–991.

Rousseeuw, P.J. (1984). Least median of squares regression. *J. Amer. Stat. Assoc.* 79, 871–880.

Rousseeuw, P.J. (1987). *Robust Regression and Outlier Detection*. Wiley, New York.

Steele, J.M. and W.L. Steiger (1986). Algorithms and complexity for least median of squares regression. *Discrete Appl. Math.* 14, 93–100.

Witkin, A.P. (1981). Recovering surface shape and orientation from texture. *Artificial Intelligence* 17, 17–45.