LOCATING TEXTURE AND OBJECT BOUNDARIES

Donald Geman [1]
Department of Mathematics and Statistics
University of Massachusetts
Amherst, Massachusetts 01003/USA

Stuart Geman [2]
Division of Applied Mathematics
Brown University
Providence, Rhode Island 02912/USA

Christine Graffigne[2]
Division of Applied Mathematics
Brown University
Providence, Rhode Island 02912/USA

ABSTRACT. Two models are given for the extraction of boundaries in digital images, one for discriminating textures and the other for discriminating objects. In both cases a Markov random field is constructed as a prior distribution over intensities (observed) and labels (unobserved); the labels are either the texture types or boundary indicators. The posterior distribution, i.e. the conditional distribution over the labels given the intensities, is then analyzed by a Monte-Carlo algorithm called stochastic relaxation. The final labeling corresponds to a local maximum of the posterior likelihood.

INTRODUCTION. A fundamental problem in computer vision (automated perception) is the semantical analysis of a three-dimensional scene based on a two-dimensional intensity image. Most techniques for recognition and labeling are rather ad hoc and often restricted to special domains, for instance industrial automation. A major goal of our past ([9], [10], [11]) and present research has been to develop a unified family of stochastic models and algorithms, which are mathematically coherent and which can be crafted to specific tasks in low and "middle" level image processing, such as filtering and de-convolution, tomographic reconstruction, object identification, and boundary and texture analysis.

It is a general perception that "segmentation", the decomposition of the intensity image into sets of pixels which correspond to the objects and regions in the underlying three-dimensional scene, is an indispensable step in image understanding. In the usual paradigm, segmentation would be followed by the imposition of a relational structure among the segments, and a final description would result from "matching" these data structures against stored representations of real-world entities. In the future we hope to extend our approach to models which are fully hierarchical, with variables ranging from raw intensity

[1] Research partially supported by Office of Naval Research contrac N00014-86-K-0027 and National Science Foundation grant DMS-8401927.

[2] Research partially supported by Office of Naval Research contrac N00014-86-0037, Army Research Office contract DAAG29-83-K-0116, and National Science Foundation grant DMS-8352087.

In: Pattern Recognition Theory and Application. Ed. P. Devijver. NATO ASI Series. Springer-Verlag, Heidelberg, 1986.

data to stored models in the form of object categories and templates, such as letters. Thus, for example, the extraction of object boundaries might be guided by pending interpretations as well as general expectations about boundary geometry, as in the current work.

We are going to consider two problems which are closely related to segmentation but useful in their own right. One is texture discrimination: the data is a grey-level image consisting of several textured regions, such as grass, wood, plastic, etc., whose maximum number and possible types are known; the goal is to classify the pixels. This problem is more difficult than texture identification, in which we are presented with only one texture from a given list. Discrimination is complicated by the absence of information about the size or shape of the regions. Of course in both cases we assume models for the possible textures have been previously constructed. There are a number of applications for such algorithms. For example, regions in remotely-sensed images which correspond to ground-cover classes often have a textured quality and cannot be distinguished by methods based solely on shading, such as edge detectors and clustering algorithms. Another application is to wafer inspection: low magnification views of memory arrays appear as highly structured textures, and other geometries have a characteristic, but random, graining.

The second problem is to isolate locations in a digital image which correspond to physical discontinuities in the three-dimensional scene. These discontinuities may correspond to depth (object boundaries), surface gradients (changes in shape), or other macroscopic factors. We refer to these as "boundaries" or "macro-edges" to distinguish them from "edges" or "micro-edges" which refer to basically all significant intensity changes, whether due to noise, digitization, lighting, object boundaries, texture, etc. This crucial distinction will be amplified below. There are many approaches to boundary detection, depending for example on the degree of a priori information that is utilized and the scale of the objects (tumors, windows, houses, lakes). The detection may be "interpretation-guided" or simply the output of a general purpose algorithm based on expectations about boundary behavior, in regard to curvature, connectedness, etc. Digital boundaries tend to be quite noisy, depending on the lighting, resolution, and other factors. It is commonplace for a relatively "sharp" boundary segment in the original scene to be associated in the digital image with a band of pixels over which one object or sub-object slowly gives way to another. In addition, pronounced gradients may appear due to noise or surface irregularities or, conversely, may disappear along actual object boundaries.

Finally, we shall not be concerned here with segmentation per se, which is often superfluous. Further processing may proceed based solely on the detected boundaries, even if the connected components do not produce a "good" segmentation. Applications include stereo-mapping, automated navigation, and the detection of natural and man-made structures (such as roads, rivers, lakes and crop boundaries) in remotely-sensed images.

MODELS FOR TEXTURE AND TEXTURE DISCRIMINATION. The data is a grey level image consisting of at most M textures of known type; we wish to label each pixel as 1,2,..., or M to indicate its class. We refer the reader to [5] for an overview of approaches to texture modeling. In addition, see the paper of Oja [17] in this volume for an approach based on co-occurrence matrices. More similar to our approach are those of Elliott and Derin [8] and Cohen and Cooper [4]. What follows is a sketch of our model. All the details will appear in a forthcoming paper [12]. (The same remark applies to the boundary model.) As for the stochastic relaxation algorithms, the reader can find a complete discussion in [9]; see also the excellent review of simulated annealing by Aarts and van Laarhoven [1].

Fix a texture type, say l, and let S denote the sites of the $N \times N$ lattice $\{(i,j) : 1 \leq i,j \leq N\}$. Our model is a Gibbs distribution based only on pair interactions, in fact just grey level differences. The parameters are the interaction weights, say $\theta_1,...,\theta_I$, where

I denotes the number of pair-bonds under consideration. The sign of θ_i will indicate the promotion of similarity or dissimilarity for bond type i.

The distribution of the grey levels $X = \{X_s,\ s \in S\}$, for the texture type l, is then

$$P(X = x) = \frac{exp\{-U^{(l)}(x)\}}{Z^{(l)}}$$

where $Z^{(l)}$ is the usual normalizing constant

$$Z^{(l)} = \sum_x exp\{-U^{(l)}(x)\}$$

and the "energy function" $U^{(l)}$ is given by

$$(1) \qquad U^{(l)}(x) = -\sum_{i=1}^{I} \sum_{<s,t>_i} \theta_i^{(l)} \Phi(x_s - x_t), \quad \Phi(\Delta) \doteq \left(1 + (\frac{|\Delta|}{\delta})^2\right)^{-1}$$

The notation $< s, t >_i$ indicates a pair of pixels of bond type i; for example, with I = 6, the types are shown in Figure 1. They correspond to the first and second nearest neighbors in the vertical and horizontal directions and nearest neighbors in the two diagonal directions. We refer again to [11] for a discussion of the choice of the potential function Φ. The basic idea is to choose a function of $(x_s - x_t)$ that is monotone and bounded in $|x_s - x_t|$. Monotonicity allows us to selectively encourage similarity or dissimilarity, whereas boundedness helps to accommodate the occasional, random, exception to an expected bond relation. Having specified the form of our model, each texture type, l, is identified with an I-tuple of parameters $\theta_1^{(l)}, \ldots, \theta_I^{(l)}$.

PLACE FIGURE 1 HERE

The parameters are estimated from samples of the textures by the method of maximum pseudo-likelihood ([2], [3]), a powerful estimation technique designed for multiplicative parameters in local Gibbs distributions. It has the advantage over maximum likelihood of circumventing the partition function $Z^{(l)}$ which is entirely intractable. The least-squares method proposed recently in [8] and [18] is not feasible in this setting due to large number of possible grey levels: it is unlikely to see any fixed neighbor configuration duplicated in the sample.

Having modeled the M textures, we now construct a composite Markov random field which accounts for both texture labels, say $X^L = \{X_s^L,\ s \in S\}$ and grey levels $X^P = \{X_s^P,\ s \in S\}$. The joint distribution is

$$(2) \qquad P(X^P = x^P, X^L = x^L) = \frac{exp\{-U_1(x^P, x^L) - U_2(x^L)\}}{Z}$$

in which U_2 promotes label bonding (we expect the textures to appear in patches rather than interspersed) and U_1 specifies the interaction between labels and intensities. Specifically, we employ a simple Ising-type potential for the labels:

$$U_2(x^L) = -\beta \sum_{[s,t]} 1_{x_s^L = x_t^L} + \sum_{s \in S} w(x_s^L), \quad \beta > 0 \tag{3}$$

Here β determines the degree of clustering, $[s,t]$ indicates a pair of nearest horizontal or vertical neighbors, and $w(\cdot)$ is adjusted to eliminate bias in the label probabilities.

To describe the interaction between labels and pixels we introduce the symbols $\tau_1, \tau_2, \ldots, \tau_I$ to represent the lattice vectors associated with the I pair-bonds (Figure 1). Thus s and $s + \tau_i$ are neighbors, constituting a pair with bond type i. The interaction is then given in terms of pixel-based contributions,

$$H(x^P, l, s) \doteq -\sum_{i=1}^{I} \theta_i^{(l)} \left\{ \Phi(x_s^P - x_{s+\tau_i}^P) + \Phi(x_s^P - x_{s-\tau_i}^P) \right\} \tag{4}$$

and local sums of these called block-based contributions,

$$Z(x^P, l, s) \doteq \frac{1}{z} \sum_{t \in N_s} H(x^P, l, t). \tag{5}$$

Here, N_s is a block of sites centered at s (5 by 5 in all of our experiments), and the constant z is adjusted so that the sum of all block-based contributions reduces to $U^{(l)}$ (see(1)):

$$U^{(l)}(x^P) = \sum_{s \in S} Z(x^P, l, s) \tag{6}$$

This amounts to ensuring that each pair-bond appears exactly once ($z = 50$, for example, when N_s is 5 by 5). More or less obvious adjustments are made at boundaries. In terms of (4) and (5), the "interaction energy", $U_1(x^P, x^L)$, is written

$$U_1(x^P, x^L) = \sum_{s \in S} Z(x^P, x_s^L, s). \tag{7}$$

Because of (6), the model is consistent with (1) for homogeneous textures, $X_s^L = l, \ s \in S$. The idea is that each local texture label, X_s^L, is influenced by the pixel grey levels in a neighborhood of s.

Let us briefly examine the local characteristics of the field, specifically the conditional distributions for the labels given all the intensity data and the values of the neighboring labels. The updating mechanism and bias correction will then be more evident. (The actual neighborhoods of the Markov random field corresponding to (2) can be easily inferred from (3) and (7).) The log odds of texture type k to type j is

$$log \left\{ \frac{P(X_r^L = k | X_s^L = x_s^L, \ s \neq r \ ; \ X_s^P = x_s^P, \ s \in S)}{P(X_r^L = j | X_s^L = x_s^L, \ s \neq r \ ; \ X_s^P = x_s^P, \ s \in S)} \right\}$$

$$= Z(x^P, j, r) - Z(x^P, k, r) + \beta \sum_{t:[t,r]} (1_{x_t^L = k} - 1_{x_t^L = j}) + \omega(j) - \omega(k)$$

$$= \frac{1}{z}\sum_{i=1}^{I}\sum_{s\in N_r}(\theta_i^{(k)}-\theta_i^{(j)})\left\{\Phi(x_s^P - x_{s+\tau_i}^P)+\Phi(x_s^P - x_{s-\tau_i}^P)\right\} + \beta\sum_{t:[t,r]}(1_{x_t^L=k} - 1_{x_t^L=j}) + \omega(j)-\omega(k)$$

The first term imposes fidelity to the "data" x_s^P, and the second bonds the labels. The efficacy of the model depends on the extent to which the first term separates the two types k and j, which can be assessed by plotting histograms for the values of this quantity both for pure k and pure j data. A clean separation of the histograms signifies a good discriminator. However, since we are looking at log odds, we insist that the histograms straddle the origin, with positive (resp. negative) values associated with type k (resp. j). The function $w(\cdot)$ makes this adjustment.

A MODEL FOR OBJECT BOUNDARIES. The process consists of grey levels $X^P = \{X_s^P\}$ as above, and boundary variables $X^B = \{X_s^B,\ s \in S^B\}$, which are binary, and indicate the presence or absence of a boundary at site $s \in S^B$. These sites are indicated in Figure 2, together with those of the "micro-edges", which are marked as lines. The boundary variables interact directly with grey level differences and with the micro-edges, which are binary variables determined by the data. We have chosen to model the micro-edges deterministically because, whereas their physical causes may involve many random factors, their placement in the digital image is rather unambiguous. The situation for boundaries is somewhat the opposite: as the images reveal, there may be many equally plausible representations of the same boundary. By the way, it is for this same reason that Bayesian algorithms based on misclassification rate are unsuitable for boundary analysis; the output will generally lack the fine structure we expect of boundary configurations. Placement decisions cannot be based on the data alone; other pending labels must be considered. Consequently we favor the use of MAP estimation for boundary placement.

PLACE FIGURE 2 HERE

The literature abounds with papers on edge and boundary detection, and the subject is currently active. A common approach, rather different from ours, is to locate the zero-crossings of a differential operator, e.g. the Laplacian. In regard to the distinction between edges and boundaries, and the placement of the corresponding sites, our outlook is the same as that of Hanson and Riseman [13]; however their approach is rule-based and deterministic, via relaxation labeling. Finally, a special class of Markov fields is utilized in Devijver [7], in which fast algorithms are developed for image segmentation. See [6] and [14] for reviews.

Given the data x_s^P, $s \in S$ and threshold parameters $T_1 < T_2$, we define the edge array as follows: we declare an edge between an adjacent pair of pixels s and t, denoted e(s,t)=1, if either i) the intensity difference $|x_s^P - x_t^P|$ exceeds T_2, or ii) the difference exceeds T_1 and the difference across one of the six neighboring edge sites (see Figure 3) exceeds T_2;

otherwise e(s,t)=0.

PLACE FIGURE 3 HERE

The joint distribution between intensities and boundaries is given by

$$P(X^P = x^P, X^B = x^B) = \frac{exp\{-U_1(x^P, x^B) - U_2(x^B)\}}{Z} \tag{8}$$

where x^P and x^B indicate grey level and boundary configurations. The first term U_1 provides "boundary seeding" and the second term provides boundary organization in accordance with our geometrical and topological expectations. The seeding is based on contrast and continuation; specifically,

$$U_1 = \theta_1 \sum_{<s,t>} x_s^B x_t^B \Psi(\Delta_{st}) + \theta_2 \sum_{s \in S^B} (x_s^B - \eta_s)^2, \quad \theta_1 < 0, \quad \theta_2 > 0$$

where $< s, t >$ denotes a pair of adjacent horizontal or vertical boundary sites, Δ_{st} is the intensity difference across the edge site associated with s,t and $\Psi(u) = u^4/(C + u^4)$. The second term correlates x_s^B with an index of connectedness η_s which is 1 if there is a "string" of four connected edges with an interior junction at s, and 0 otherwise. The matrix η_s need only be computed once and there is a simple formula.

The second term in (8) is intended to "organize" the boundary "candidates" associated with the low energy states of U_1. Of course these two processes, seeding and organization, are entirely cooperative: low contrast segments may survive if sufficiently well organized and, conversely, many unstructured segments are eliminated by the "smoothing" effect of U_2.

To reduce curvature and inhibit multiple representations we penalize the simultaneous occurrence of boundaries over any of the cliques depicted in Figure 4a, together with their rotations by $\pi/2$. Let $\mathcal{C}_1$ denote this class of cliques; obviously the choice will depend on information about the-imagery. Thus

$$U_2 = \theta_3 \sum_{C \in \mathcal{C}_1} \prod_{s \in C} x_s^B - \theta_4 \sum_{C \in \mathcal{C}_2} \mu(x_s^B, \ s \in C), \quad \theta_3, \theta_4 > 0 \tag{9}$$

The other term in (9) is based on the cliques in Figure 4b. There are 32 possible configurations. The potential function μ down-weights isolated boundaries and endings and favors continuation and other "proper" configurations. One can obtain far better looking boundaries by fitting curves in a post-processing step. We have not done so.

PLACE FIGURE 4 HERE

The parameters $\theta_1, \theta_2, \theta_3, \theta_4$ are not estimated from the data. Rather, they are de-

termined by a method we have called "reparameterization" and which is developed in [12]. Basically the idea is this: We delineate a series of local "situations" in which we can conveniently quantify our a priori expectations about proper boundary behavior. Each situation leads to a constraint, typically of the form

$$a_1\theta_1 + a_2\theta_2 + a_3\theta_3 + a_4\theta_4 \leq 0$$

where the constants $a_1, ..., a_4$ may depend on "new parameters" such as the minimum contrast between uniform "object" and "background" at which "detection" is prefered. The intersection of the constraints determines a reasonably small subset of the full parameter space. Of course some "fine-tuning" is still necessary.

ALGORITHMS. In both models the posterior distribution is simply the conditional distribution of the unobserved variables (texture labels and boundary indicators) given the observed variables, i.e the grey levels. Thus, in the texture case,

$$P(X^L = x^L | X^P = x^P) = \frac{exp\{-U_1(x^P, x^L) - U_2(x^L)\}}{\sum_{x^L} exp\{-U_1(x^P, x^L) - U_2(x^L)\}}$$

and similarly for the object-boundary model. The MAP estimator, which is the most likely labeling given the data, is the Bayes rule corresponding to the zero-one loss function $L(x^L, \hat{x}^L) = 1$ if $x^L = \hat{x}^L$ and $= 0$ otherwise. We find this estimator effective in both cases, although any other method which utilizes spatial information could be equally effective. We refer the reader to Ripley [19] for an excellent discussion of the role of spatial context and performance evaluation for statistical image models. There is considerable skepticism about the desirability of a zero-one loss function; see Besag [3], Devijver [7], and Marroquin et al [15]. Two standard criticisms, at least in the context of filtering, de-convolution, and surface reconstruction, are that MAP is too "global", leading to "over-smoothing" or perhaps gross mislabeling, and that it is impractical due to excessive computation. However, pixel-based error measures, as advocated for example in [15] are simply too local for the tasks at hand. As for computational demands, the annealing and other relaxation algorithms are indeed intensive, but too much concern with CPU times can deter progress. Software engineers know that it is often possible to achieve orders-of-magnitude speed-ups by some modest compromises and reworkings when dedicating a general purpose algorithm to a specific task. Besides, advances in hardware are systematically underestimated. Indeed, it was announced at this very conference (Murray et al [16]), that experiments in [9] requiring several hours of VAX time were reproduced in less than a minute on the DAP; the authors even speculate about real-time stochastic relaxation.

In the case of the texture labeling, the processing is along now familiar lines: visit the label sites in a raster-scan fashion, replacing the current values by samples selected from the conditional distribution for the label for that site given the (current) neighboring labels and the data. In addition, a control parameter corresponding to "temperature" is reduced each full sweep (=iteration). The effect is to drive the configurations so generated towards the mode. We refer to this as single-site simulated annealing (SSSA).

The boundary-finding algorithm consists of three distinct steps. First, select threshold values and determine the corresponding location of the micro-edges. (This step is nearly instantaneous.) Next, use SSSA (initialized with $X^B \equiv 0$) to locate a low energy state of the posterior distribution. This requires several hundred sweeps and results in a reasonable, but "flawed" boundary (See the upper right panels in Figures 9, 10, and 11). Finally, we take this state as the starting point of an iterative improvement algorithm. Successively visit each cross-shaped set of five boundary sites and choose the most likely value for these five sites given the remaining sites and the data. Since we are successively maximizing

the posterior energy function in five coordinates simultaneously, each such visit can only increase the posterior likelihood. We have refered to this as "0-temperature" reconstruction; Besag [3] calls it ICM, for "iterated conditional modes." This procedure converges in about three or four sweeps and is very sensitive to the starting point, in this case the good one obtained by SSSA. The second and third steps require about equal processing time.

EXPERIMENTS. Three experiments were done on texture discrimination, based on two images with two textures each and one with four. There are four textures involved: wood, plastic, carpet, and cloth. As mentioned above, the parameters were estimated from the pure types using maximum pseudo-likelihood. The experiments required about 150 iterations of SSSA, initialized by assuming no label-label interactions ($beta = 0$ in (3)) and then assigning to each label, X_s^P, its most likely value. There was no pre- or post-processing. In particular, no effort was make to "clean-up" the boundaries, expecting smooth transitions. The results are shown in Figures 5, 6, and 7; these correspond to i) wood on plastic, ii) carpet on plastic, and iii) wood, carpet, and cloth on plastic background. In each figure, the left panel is the textured scene, and the right panel shows the segmentation, with texture labels coded by grey level. It is interesting to note that the grey level histograms of the four textures are very similar (Figure 8); in particular, discrimination based on shading alone is virtually impossible. We also mention that samples generated from the model do not resemble the textures very well. This illustrates the fact that the utility of Markov field models does not depend on their capacity for simulating real-world imagery.

There are also three experiments on object boundary extraction, corresponding to three test images: two scenes constructed from tinkertoys and an outdoor scene provided by the computer vision group at the University of Massachusetts. In Figures 9 and 10 the lower left panel is the original scene, the upper left panel shows the deterministic placement of "micro-edges", the upper right panel shows the result of SSSA, and the bottom right panel, the final product, is the result of five-site iterative improvement, initialized with the output of SSSA. The stick scene (Figure 9) is 64x64 and illustrates the ambiguity in digital boundaries. The "cart" (Figure 10) is 110x110; we were quite pleased to obtain such straight lines without post-processing. Finally, the house scene (Figure 11) is 256x256 and obviously more varied than the others. The upper left panel is the original scene, the upper right is the result of SSSA, and the bottom panel shows the final placements of boundaries. We re-emphasize that selecting the parameters is arduous and still somewhat ad hoc; we would of course prefer a simple, data-driven method based on solid statistical principles. Apart from the threshold values which depend on the grey level histogram, the same parameters were used in all three experiments.

PLACE FIGURES 5 THROUGH 11 HERE

REFERENCES

1. E. Aarts and P. van Laarhoven, "Simulated annealing: a pedestrian review of the theory and some applications," NATO Advanced Study Institute on Pattern Recognition: Theory and Applications, Spa, Belgium, June 1986.

2. J. Besag, "Spatial interaction and the statistical analysis of lattice systems" (with discussion), J. Royal Statist. Soc., Series B, 36, 192-236, 1974.

3. J. Besag, "On the statistical analysis of dirty pictures," J. Royal Statist. Soc., Series B, 1986.

4. F.S. Cohen and D.B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields," IEEE Trans. Pattern Anal. Machine Intell., to appear.

5. G.R. Cross and A.K. Jain, "Markov random field texture models," IEEE Trans. Pattern Anal. Machine Intell., PAMI-5, 25-40, 1983.

6. L.S. Davis, "A survey of edge detection techniques," Comput. Graphics Image Processing, 4, 248-270, 1975.

7. P.A. Devijver, "Hidden Markov models for speech and images," Nato Advanced Study Institute on Pattern Recognition: Theory and Applications, Spa, Belgium, June 1986.

8. H. Elliott and H. Derin, "Modelling and segmentation of noisy and textured images using Gibbs random fields," IEEE Trans. Pattern Anal. Machine Intell., to appear.

9. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," IEEE Trans. Pattern Anal. Machine Intell., 6, 721-741, 1984.

10. D. Geman and S. Geman, "Bayesian image analysis," in Disordered Systems and Biological Organization, Springer-Verlag, Berlin, 1986.

11. S. Geman and D.E. McClure, "Bayesian image analysis: an application to single photon emission tomography," 1985.

12. D. Geman, S. Geman, and D.E. McClure, "Markov random field image models and their applications," invited paper, Annals of Statistics, in preparation.

13. A.R. Hanson and E.M. Riseman, "Segmentation of natural scenes," in Computer Vision Systems, Academic Press, New York, 1978.

14. D. Marr and E. Hildreth, "Theory of edge detectors," Proc. Royal Soc. B, 207, 187-207, 1980.

15. J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," artif. intell. lab. TECH. REPORT, M.I.T., 1985.

16. D.W. Murray, A.Kashko, and H. Buxton, "A parallel approach to the picture restoration algorithm of Geman and Geman on an SIMD machine," preprint, 1986.

17. E. Oja, "Texture subspaces," NATO Advanced Study Institute: Theory and Applications, Spa, Belgium, June 1986.

18. A. Possolo, "Estimation of binary Markov random fields," preprint, 1986.

19. B.D. Ripley, "Statistics, images, and pattern recognition," Canadian J. of Statist., 14, 83-111, 1986.

Figure Legends

Figure 1. Pair-bonds for texture model.

Figure 2. Pixel, "micro-edge", and boundary sites.

Figure 3. Pixel ($\circ$) and edge ($|$,—) sites that determine the state of edge $< s, t >$.

Figure 4. Clique types for boundary potentials.

Figure 5. Wood on plastic background.

Figure 6. Carpet on plastic background.

Figure 7. Wood, carpet, and cloth on plastic background.

Figure 8. Grey-level histograms.

Figure 9. Stick scene.

Figure 10. "Cart".

Figure 11. House scene.

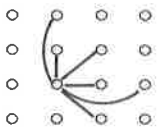


Figure 1

∘ Pixels

|,— Edge Sites

+ Boundary Sites

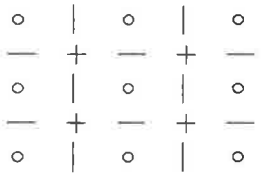


Figure 2

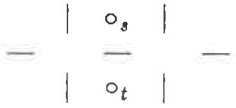


Figure 3

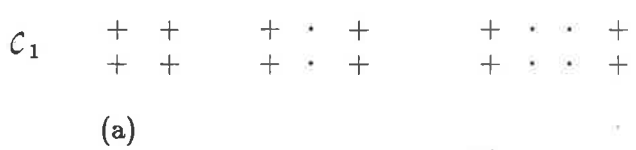


Figure 4

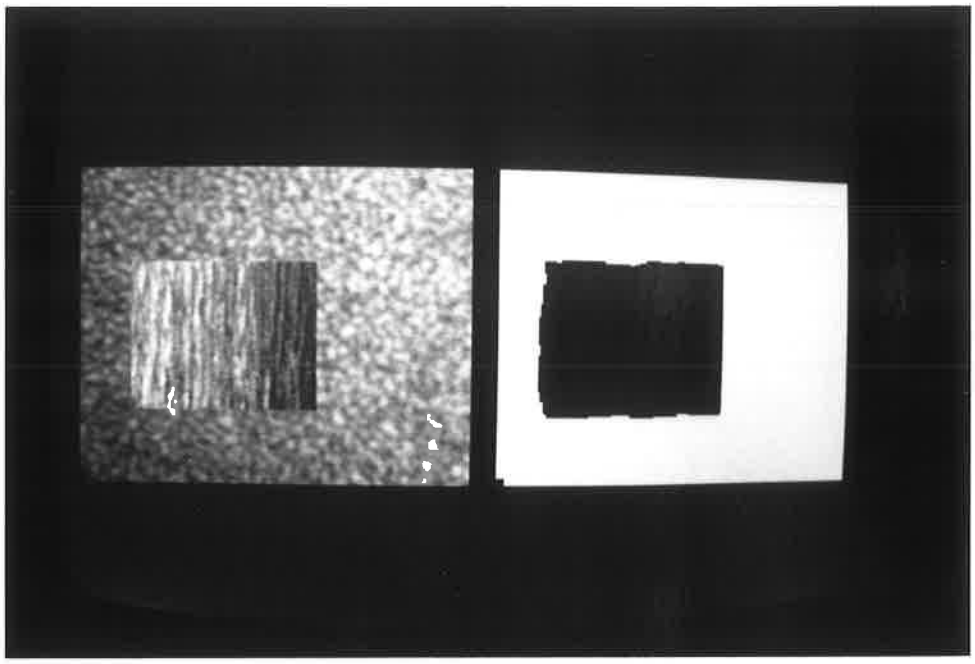


Figure 5

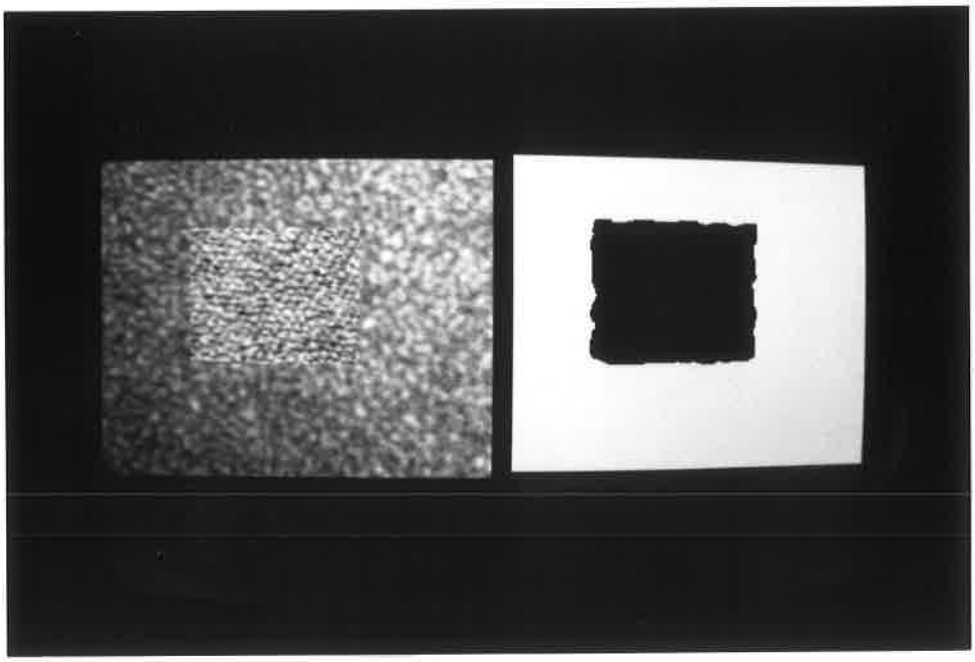


Figure 6

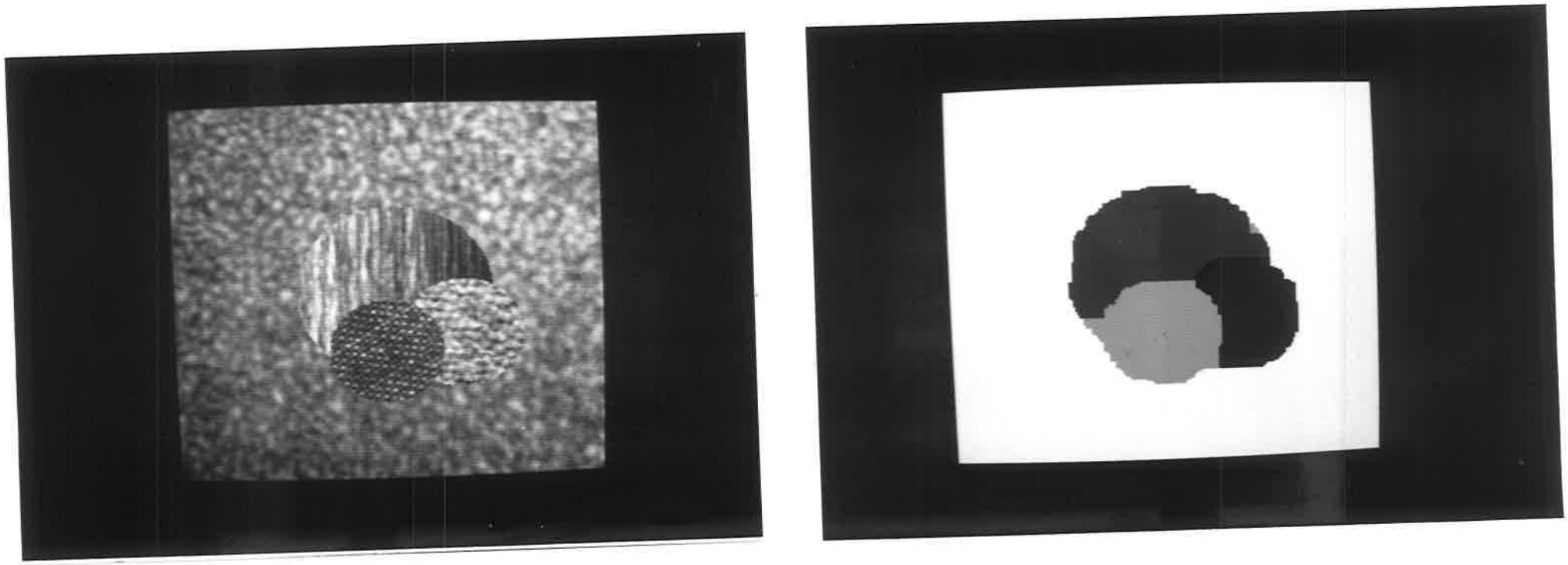
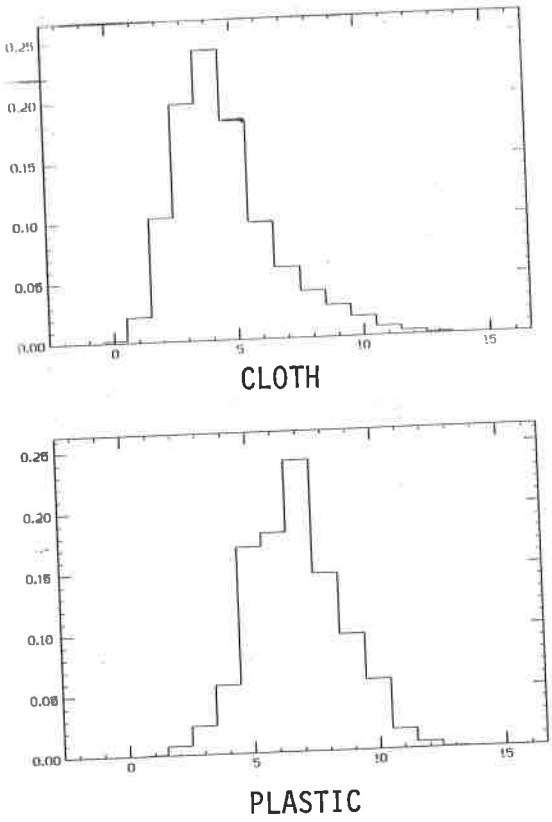
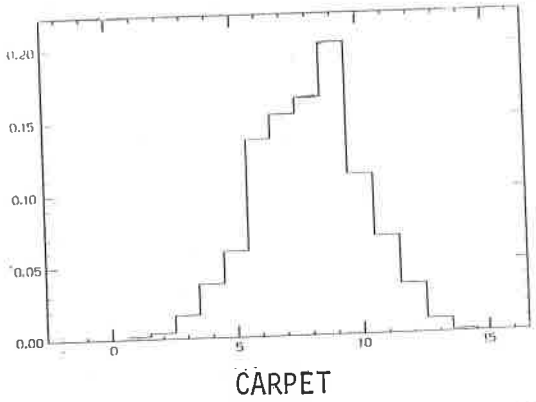


Figure 7


0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
CLOTH


0.20
0.15
0.10
0.05
0.00
0
5
10
15
CARPET

0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
PLASTIC

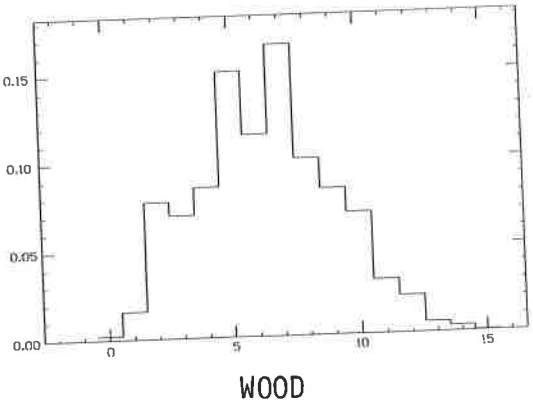

0.15
0.10
0.05
0.00
0
5
10
15
WOOD

Figure 8

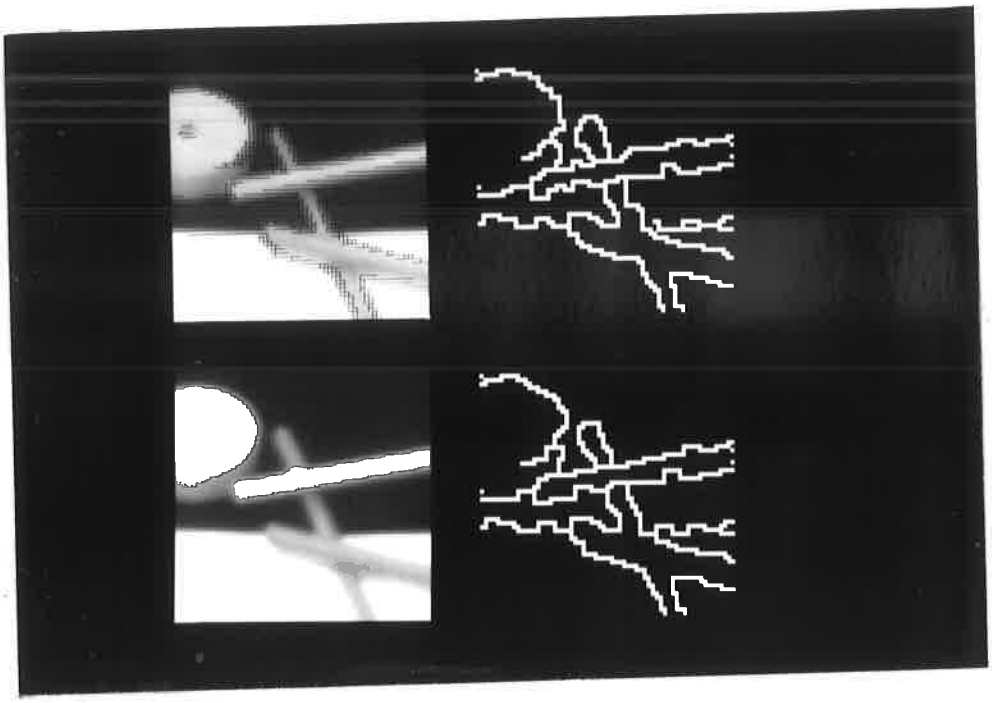


Figure 9

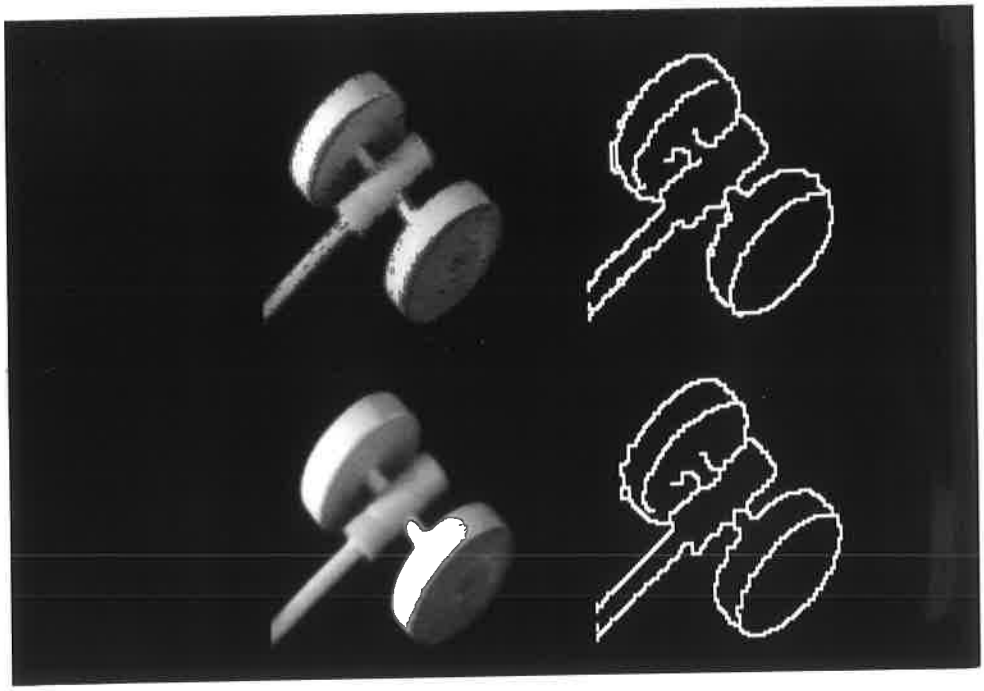


Figure 10

Figure 11