TR－A－0113

Simplifying Discontinuity Detection with an Eye on Recognition

Ed Gamble

1991. 4. 23

ATR視聴覚機構研究所

〒619-02 京都府相楽郡精華町乾谷 ☎07749-5-1411

ATR Auditory and Visual Perception Research Laboratories

Inuidani,Sanpeidani, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1411
Facsimile: +81-7749-5-1408
Telex: 5452-516 ATR J

Contents

Abstract

The essence of our approach is to address the important problem of discontinuity detection within the context of the overall visual recognition problem[1]. Awareness of the capabilities and expectations embodied in recognition algorithms and of the dominant noise process in the surface properties computed by some early vision algorithms greatly simplifies the detection of discontinuity. In particular, we describe the characteristic "displacement" errors, that stereo and optical-flow algorithms produce near object boundaries and we suggest that the detected discontinuities, in light of these errors, must be restricted to a subset of intensity edges. This restriction simplifies discontinuity detection and is valid under certain assumptions which we describe.

We have detected discontinuities in depth and in the magnitude of optical flow for a variety of natural images by combining intensity edges and surface property data computed with early vision algorithms. The integration of surface properties is formulated as an optimization problem derived from a Markov random field. A massively parallel, stochastic relaxation algorithm for solution of these optimization problems is described.

1 Prelude

The detection of discontinuities is an important problem in computer vision. A discontinuity represents those locations in an image where some property of the surface in the imaged scene

[1]Portions of this work were performed in the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. A revised portion of this paper will appear in the IEEE CVPR 1991 conference proceedings. The original submission to CVPR '91 was dated 19 November 1990.

changes abruptly. Such changes are important because, under some general assumptions, they correspond for example to boundaries between objects (or subparts of an object) and to facets of an object's surface. Consequently, detection of discontinuities may facilitate segmentation of an image into objects and possibly to symbolic representations of an image's components. In general, a discontinuity may arise due to a change in any property of a surface. Common examples of discontinuities include depth discontinuities from the reconstruction of a surface from stereo depth and motion discontinuities from the reconstruction of a flow-field from optical-flow data.

The surface properties computed by stereo and optical-flow algorithms are fraught with errors. These errors invariedably manifest themselves at first order discontinuities because surface properties are computed with algorithms which assume that surfaces are continuous. However, discontinuities are precisely those location where surface continuity is violated. These errors are so pervasive and have such a distinctive signature that analysis of the error can be employed as a discontinuity detector[1, 2]. Somewhat surprisingly therefore, few surface reconstruction algorithms have adopted a noise model consistent with this pervasive noise in the input data to surface reconstruction.

The detection of discontinuities is difficult because the problem of surface reconstruction is an example of an ill-posed problem[3]. Surface reconstruction algorithms must disambiguate among the multitude of surfaces consistent with the noisy and sometimes sparse surface data provided as input. There is no apriori way to select among these surfaces; assumptions are required[4]. Assumptions are based upon known or otherwise desired properties of surfaces, such as smoothness, and also upon consideration of the noise processes that degrade the input data[5]. Ever more elaborate schemes simultaneously smooth surfaces, detect discontinuities, and smooth the discontinuities themselves[6, 7] and operate over multiple scales[8] for all orders of surface discontinuity[9]. Although surface reconstruction is interesting in its own right, is it possible that these elaborate surface reconstruction algorithms are unnecessary when object recognition is considered the ultimate goal of vision processing?

The essence of our approach is to address the problem of discontinuity detection *within the context of the overall visual recognition problem.* Awareness of the capabilities and ex-

pectations embodied in recognition algorithms greatly simplifies the discontinuity detection. Essentially, the apriori knowledge about recognition imposes additional constraints on the surface reconstruction problem. Similarly an awareness of the pervasive noise process in the input data further constrains the surface reconstruction problem. On the next few pages we describe the characteristics of surface properties produced by early vision algorithms and also describe the capabilities of recognition algorithms. These two issues provide the basis upon which our formulation is based.

1.1 Background

Vision systems for unrestricted environments generally include an image processing stage known as early vision[10]. Early vision algorithms compute the surface properties, such as depth, optical flow, and texture of the three-dimensional scene throughout the image. These computations are based on one or more two-dimensional images that are devoid of direct information regarding the physical properties of surfaces. The solutions to these early vision problems require assumptions regarding smoothness[11] because, as in the the case of surface reconstruction, early vision problems are generally ill-posed[12]. Since the surface properties computed by these early vision algorithms are the inputs to the surface reconstruction algorithm, an examination of the dominant noise in the surface properties is of critical importance.

In order to examine the noise in surface property data, we consider a typical early vision algorithm for the computation of optical flow[13]. Assume that two images exist, $E_1(\vec{r})$ and $E_2(\vec{r})$, where $\vec{r} = x\hat{x} + y\hat{y}$ is a vector in the image plane (with axes $\hat{x}$ and $\hat{y}$) and $E(\vec{r})$ is the intensity measured by the imaging device at pixel (x, y). The fundamental problem is to find the correspondence between each of the pixels in the two images. The x-y translation in image coordinates between the two corresponding pixels is the *disparity*, $\vec{d}(\vec{r})$, where $\vec{d}(\vec{r}) = d_x(\vec{r})\hat{x} + d_y(\vec{r})\hat{y}$. A solution for $\vec{d}$ is found by minimizing

$$I = \int \left\{ \phi \left[E_1(\vec{r}), E_2(\vec{r} + \vec{d}) \right] + \lambda \|(\nabla \vec{d})\|^2 \right\} d\vec{r}. \tag{1}$$

Because of the smoothing constraint, $\vec{d}(\vec{r})$ is a slowly varying function of $\vec{r}$. If $\vec{d}(\vec{r})$ is assumed

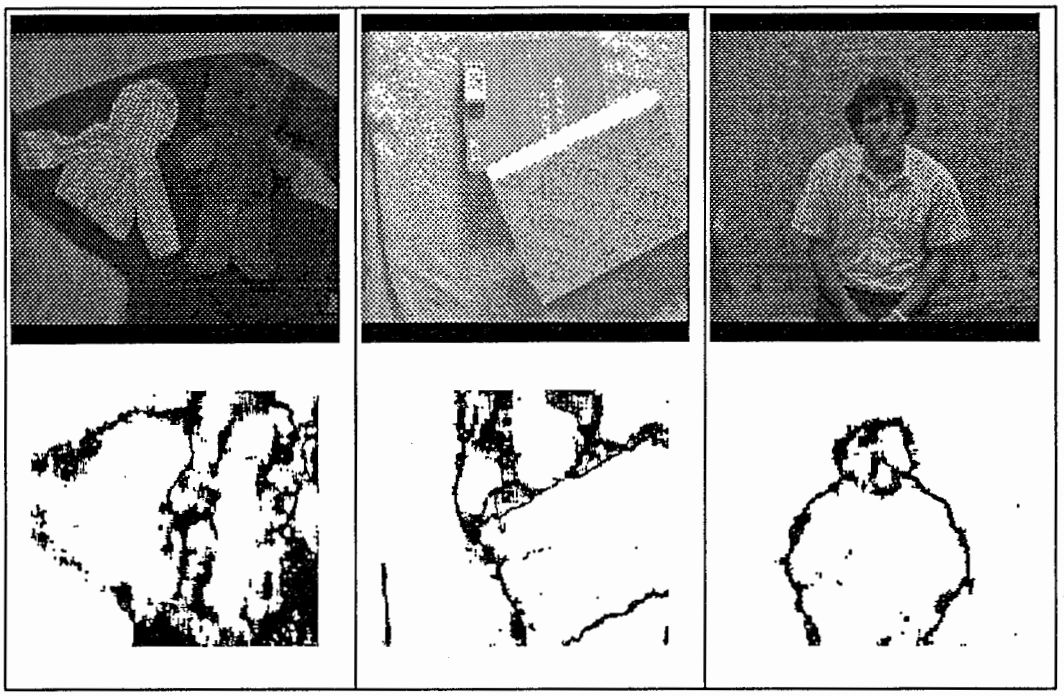


Figure 1: Examples of noise in correlation detectors. For stereo disparity, regions with multimodal correlation are shown below the left image of a stereo pair.

constant over regions of size A, then Equation 1 can be computed by finding $\vec{d}(\vec{r})$ that satisfies:

$$\max S(\vec{d}(\vec{r})) = \max \int_A \phi \left[E_1(\vec{r}), E_2(\vec{r} + \vec{d}) \right] d\vec{r}. \tag{2}$$

$S()$ is called the *match score* or *correlation function*; the computation is a correlation in $\vec{d}(\vec{r})$ across $\vec{r}$. The analysis for stereo is similar except that for the computation of $S()$, $\vec{r}$ is restricted to lie on an epipolar line.

The functional form of $S()$ yields important information about the matching process[1]. If $S()$ is multimodal then, generally, the smoothness constraint has been violated. This occurs near discontinuities in motion or depth where part of region A correlates well for, say, d_1 but another part of A correlates well at d_2. Examples of multimodal correlation functions have been presented for stereo[2] and optical flow[1]. Figure 1 contains three images; each image is accompanied by a binary map that highlights those pixels where the correlation function is multimodal. These binary maps were computed with a pixel-based stereo algorithm[2] in which $\phi()$ of Equation 1 was the square of the difference between the intensity of each pixel in

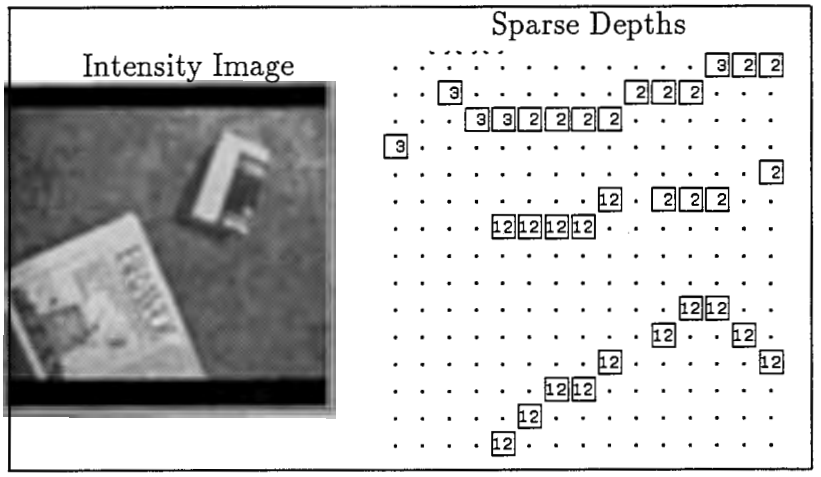


Figure 2: An example of displacement errors in surface property data. The "sparse depths" picture is a close-up of the stereo depth data from a small region at the upper corner of the "BRAVO!" newspaper.

a stereo pair. The figure shows that the multimodal regions always occur near discontinuities in depth and, in some cases, in places unrelated to discontinuities.

The multimodal behavior manifests itself in the computed disparity field as a displacement of disparity data. *Displacement* refers to one pixel near a discontinuity reporting the disparity of the pixels on the other side of the discontinuity; this displaced pixel in on the wrong side of the "actual" discontinuity. Figure 2 illustrates this displacement error; the surface with disparity 2 and 3 is corrupted by 5 pixels in the middle-left of the figure with a disparity of 12. The significance of these pervasive errors has been all but ignored in surface reconstruction and discontinuity detection algorithms[1]

The simple question is how should a surface reconstruction algorithm handle these displacement errors? Based on the disparity data only, a surface reconstruction algorithm *must* mark discontinuities as displaced from their actual location. Yet, clearly this results in an error in the reconstructed surface. In fact, for a surface reconstruction algorithm based on surface property data alone, there is no apriori method to distinguish displacement errors. Additional information is required and hence integration of visual information must be

[1]A possible explanation as to why these errors have been overlooked is that most researchers in discontinuity detection are also interested in edge detection (for example [14]). Edge detection is the process of finding discontinuities in intensity. However, because of the optics of image formation, intensity data is not plagued by displacement errors.

performed.

Early work on *intrinsic images*[15] suggested the importance of integration for reconstructing surface properties and their discontinuities. Unfortunately, at that time, neither an appropriate mathematical foundation nor a basis for integration were elucidated. Only later did a unified mathematical approach (Markov random fields) and the notion of *observable discontinuities*[16] appear for integration. Observable discontinuities are a rough estimate of the discontinuities in surface properties and they can be derived, for example, from the match scores in correlation-based early vision algorithms. However as we have illustrated in Figure 1, these match scores and consequently observable discontinuities are themselves plagued by displacement errors.

Early attempts at integration[17, 18] used intensity edges to guide the search for surface property discontinuities while simultaneously detecting discontinuities. Because neither of these attempts recognized the pervasiveness of displacement errors, both formulations suffered from needless complexity and computational inefficiency. These complexities and inefficiencies will come to light in the next sections when the formulation and results are described. Prior to this however fully framing the vision problem demands a brief discussion of the capabilities and expectations of current recognition algorithms.

Many types of recognition algorithms exist[19, 20]. However, for the discussion here, there are two key aspects of recognition algorithms: generally, they are capable of dealing with occluded contours and they explicitly discount the use of depth information[21]. Examples of model-based recognition algorithms based on 2D or 3D models from only 2D image data include constrained search[22], alignment[23], and grouping[21]. These algorithms use model and image features, such as lines, corners, and/or curves, derived from intensity edges. Variations in these features may arise due to viewpoint and illumination; one example is occlusion whereby one object obstructs the contour of another object. In practice, occlusion also may occur when an edge detector fails to identify an object's contour. The model-based recognition algorithm referenced above all account for occlusion by identifying objects when given only a portion of an object's contour.

Besides occluded contours, images may contain spurious features that do not serve to distinguish objects. Elimination of this *clutter* can dramatically reduces the algorithmic

complexity in some algorithms[24].

2 Formulation

In this chapter we exploit the information developed in the previous chapter to simplify the discontinuity detector. For the purpose of formulating the discontinuity detector the previous chapter can be summarized as follows.

I-1 The assumptions inherent in surface property algorithms used in early vision produce displacement errors near the surface property boundary. These errors are pervasive.

I-2 Model-based recognition algorithms account for occlusion of image contours. The computational complexity of a recognition algorithm is improved by a reduction in the number of irrelevant contours.

If, in addition to these items, we make explicit the fundamental assumption of passive vision algorithms:

I-3 Changes in a surface property originate intensity variations in the image,

then we have a prescription for a simple discontinuity detector.

In our formulation, the detected discontinuities in surface properties must be a *subset* of the contours that would otherwise be used for recognition. The key here is to view discontinuity detection as a labeling process whereby labels, such as "depth discontinuity," are associated with each pixel along a contour. In this way, discontinuity detection is akin to the processes of saliency[25] or perceptual grouping[21]. Both these processes classify contours or contour pixels based on various measures such as contour length, smoothness, and colinearity. The discontinuity labels serve to further label the contours. The advantages of this subtle restriction that discontinuities be a subset of image contours are significant; these advantages are detailed below.

Localization Because of the optics of image formation, the intensity variations in the image coincide with the projection of the discontinuity from the 3-D scene onto the image plane. Within certain edge detector constraints, the intensity edges derived from the

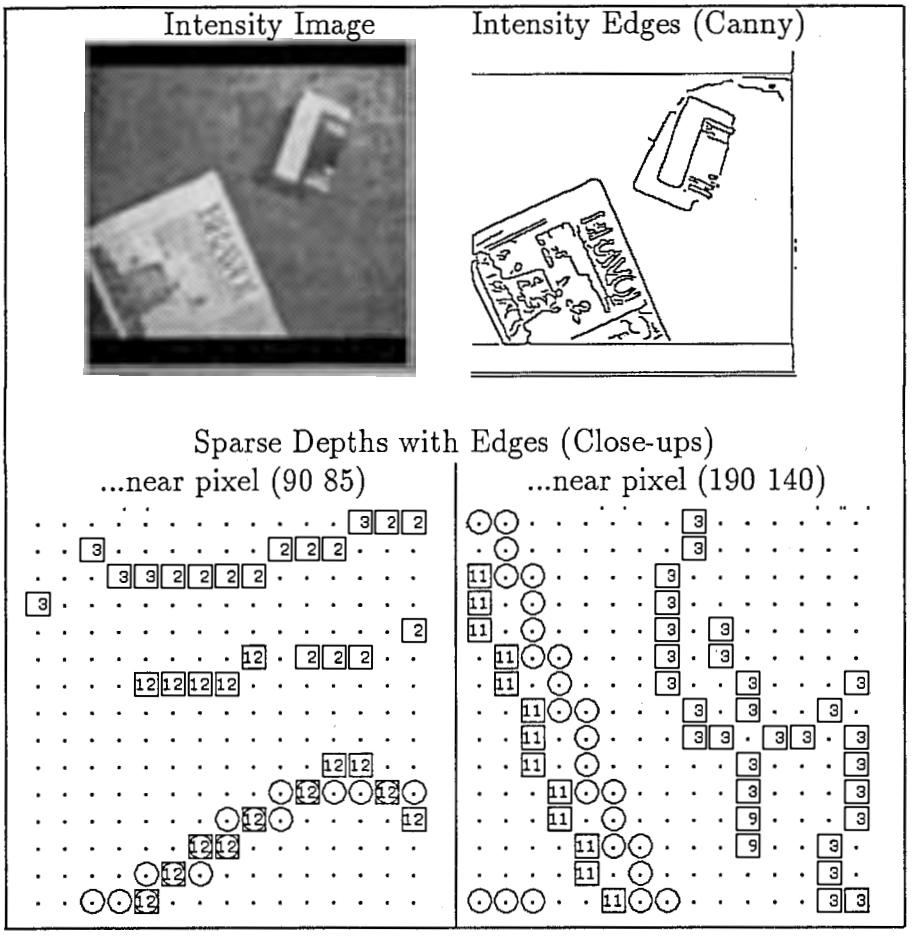


Figure 3: The intensity edges (circles) are displayed with the sparse depth data (boxed numbers). With sparse data, the intensity edges eliminate the inherent uncertainty in discontinuity location. Also the intensity edges are smoother than the depth data noise allows.

image will be accurately located whereas, as describe earlier, the surface properties are particularly noisy near discontinuities. Also because early vision algorithms often use sparse features for matching, the surface properties may be sparse and consequently discontinuity location can be poor. By restricting discontinuites to intensity edges, the discontinuity locations will be as accurate as the intensity edges. Figure 3 shows this effect of discontinuity *localization.*

Smoothness A major problem with surface reconstruction and discontinuity detection is the creation of an image model that embodies the assumption of smoothness and continuity of discontinuities. Within probabilistic formulations, many ingenious attempts have been devised to enhance the smoothness of discontinuities[5, 26, 27]. These attempts all seek to overcome the local interactions mandated by probabilistic formulations, such as Markov random fields (Mrfs). For deterministic formulations[7, 28, 29], smoothness and continuity contraints are abandoned in favor of computabilility concerns. Further, multiple scale formulations[8] and even renormalization group formulations[30] are, at some level, attempts to extend the smoothness of discontinuities beyond the local interaction regions common in these formulations. However, when discontinuities are a subset of intensity edges the smoothness of the discontinuities is comparable to the smoothness of the intensity edges. Complex machinations[2] are no longer required to impose smoothness. Compare the smoothness of the intensity edges in Figure 3 with the possible discontinuities through the sparse region or with the noisy multimodal binary map in Figure 1.

Displacement As Figure 2 illustrated, displacement errors from early vision algorithms cannot be eliminated unless additional information is provided. The restriction of discontinuities to intensity edges ensures that any surface property data that is not near an intensity edge and that differs from nearby surface property data will be considered a displacement error. The validity of this model rests upon item I-3 described at the

[2]We refer, in particular, to Mrf formulations in which the specification of clique energies for the line process poses a major problem.

beginning of this section; surface property changes originate intensity variations.[3]

Computability The three prior items: localization, smoothness and displacment, when embodied in a Markov random field, simplify the stochastic relaxation algorithm. The Mrf optimization function does not require complex neighborhood interactions to impose smoothness. Furthermore, these items simplify previous deterministic formulations for surface reconstruction[8, 7, 28, 29]. Consequently, hardware implementation, applicable to real images, may be possible.

Focus The restriction of discontinuities to a subset of intensity edges emphasizes detection of discontinuities over reconstruction of surfaces. With this view, the discontinuity detector can be viewed as a type of perceptual grouping[21] algorithm in which the underlying physical event producing the intensity edge is used to label the edge. Recognition is improved because discontinuities are provided, irrelevant contours are eliminated from the features for recognition and the reduced number of features favor the recognition combinatorics. These improvements are identical to benefits derived from saliency and perceptual grouping algorithms although saliency and perceptual grouping derive their improvement upon inter-relationships between contours and contours pixels rather than upon surface property discontinuities.

We now formalize the inputs and outputs for the discontinuity detection and the surface reconstruction problems. As the problem is framed eventually is a Markov random field, we also briefly discuss the parallel Gibbs sampler algorithm. Finally, the optimization function for our formulation is presented.

2.1 Representation of the Surface and its Discontinuities

We define the two-dimensional image as a *lattice*, S, where $S = \{s_1, s_2, \cdots, s_{\mathcal{N}}\}$ and s identifies a site in the two-dimensional, N x N pixel array with total size $\mathcal{N}$. Associated with

[3]Frequently, texture discontinuities and isoluminant color boundaries do not produce intensity variations at discontinuities; these types of surface properties have fundamentally different noise properties and are not treated here.

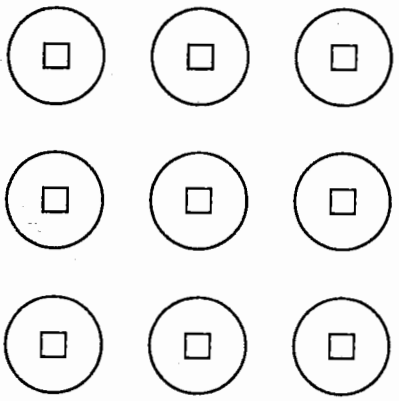


Figure 4: A multivariate field composed of a feature process (small squares) and a discontinuity process (large circle) in an 8-connected neighborhood is shown. When the discontinuity process is on, $l_s = 1$, smoothing of the feature process is inhibited between *all* sites $s' \in Q_s$ for which $l_{s'} \neq 1$.

each site is a site *state space*, Λ, with a *size*, l_X and *coordinates* from the set $\{0, 1, \cdots, l_X - 1\}$. A *field* is defined as $X = \{x_s, s \in S\}$ where $X_s \in \Lambda$.

For our surface reconstruction and discontinuity detection, X will be a multivariate field[31] with $X = \{F, L\}$, $\Lambda_X = \Lambda_F \cdot \Lambda_L$ and $l_X = l_F l_L$. The field $F = \{f_s, s \in S\}$ is the *feature* process which represents the reconstructed surface property. In general, f_s can be continuous or discrete; however, because of the discrete nature of disparity data, we choose f_s as discrete. The field $L = \{l_s, s \in S\}$ is the *discontinuity* process.[4] Following earlier work[32], the discontinuity process is a binary field with $l_L = 2$, and $l_s \in \{0, 1\}$. When $l_s = 1$, the site s is labeled as a discontinuity. These fields are illustrated in Figure 4.

The input to the surface reconstruction is comprised of two fields, $Y = \{G, E\}$. One field is the sparse surface property information, G; the other is the intensity edges, E. The surface properties are represented by two variables, g_s and γ_s for site s. As in the case of f_s, g_s is discrete. The variable γ_s encodes the sparseness of the early vision output and is defined as $\gamma_s = 1$ if input data exists at lattice site s; otherwise $\gamma_s = 0$. The intensity

[4]Historically, the discontinuity process, L, has been called the *line* process[5]. Although we maintain the label "L" for the field, we insist on the use of "discontinuity" to emphasis the fundamental differences between this and other approaches. In particular, our discontinuity process is *not* formulated on a dual lattice[5]. The dual lattice is composed of two binary processes; a horizontal and a vertical process which conceptually are located between lattice sites. These two process inhibit smoothing only between two neighbors; whereas, our discontinuity process inhibits smoothing between all neighbors as appropriate.

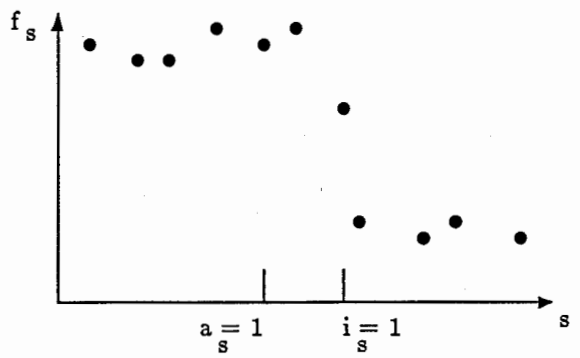


Figure 5: Displacement errors occur when the indicated discontinuity, $i_s = 1$, and the actual discontinuities, $a_s = 1$, do not coincide. The above plot shows the sparse and noisy surface property data, f_s, and the misplaced indicated discontinuity. Item I-3 at the start of this section suggests that $a_s = 1 \iff e_s = 1$.

edges, represented by the field E, are similar to the discontinuity process, L, except that when $e_s = 1$ a discontinuity may occur at lattice site s. When $e_s = 0$ the site s is strictly forbidden from marking a discontinuity with $l_s = 1$. The field E is typically computed with Canny's edge detector[33].

Figure 5 illustrates the displacement errors in the surface property data G. Let $I = \{i_s | s \in S\}$ be the *indicated* discontinuity field. $i_s = 1$ indicates that the derivative of surface property data is large at site s. Let $A = \{a_s | s \in S\}$ be the actual discontinuity field. $a_s = 1$ identifies a real, physical discontinuity in the scene that is located at s in the image. A displacement error occurs when $i_s = 0$ and $i_{s'} = 1$ for sites s and s' such that $a_s = 1$ and $s' \sim s$. (This is referred to as a displacement error at s'.) The goal of discontinuity detection is the computation of A; however, surface property data supplies I. Furthermore, because of displacement errors in the surface property data, $A \neq I$ and therefore, without an integration scheme the discontinuity detector fails by producing $L = I$ rather than $L = A$.

As described previously, our integration scheme assumes that physical discontinuities in the scene originate intensity variations in the image. Consequently, the assumption for the image is $A \subseteq E$. Our discontinuity detection algorithm eliminates the displacement errors at s' by shifting the discontinuity from s' to $\{s | e_s = 1\}$. The shift is performed by smoothing F while temporarily fixing L such that $L = E$. Subsequent to this smoothing, discontinuity detection is performed.

This two-phase algorithm, shifting then detecting, produces a significant advantage. The problem of discontinuity detection becomes akin to edge detection whereby a linear operation, smoothing, eliminates noise and then a nonlinear operation, maximizing a gradient or finding zero-crossings, locates edges. This advantage arises precisely because of our awareness of the dominant noise process in early vision algorithms, namely displacement errors, and because of the fundamental link between physical discontinuities and the scene and intensity variations in the image. Two points should be noted. First the noise process assumed by edge detectors differs significantly from the noise process that dominates surface property data, yet a similar two-phase algorithm is appropriate. And second, a two-phase algorithm can be implemented within the Markov random field (Mrf) paradigm or within paradigms significantly more efficient than Mrfs.

2.2 MRF Specification and the Parallel Gibbs Sampler

Markov random fields have been used extensively for image analysis[34]. In this section we describe a parallel stochastic relaxation algorithm that significantly improves the computational performance when the site state space size is large. Rather then describing the Mrf formulation in detail we assume some familiarity with Mrfs and present just the aspects related to the parallel implementation.

The degree to which an Mrf can be made parallel depends on the neighborhood system Q of the lattice sites S. This Mrf *neighborhood*, Q, is defined as $Q = \{Q_s, s \in S\}$ where Q_s satisfies: 1) $s \notin Q_s$, and 2) $s \in Q_r \iff r \in Q_s$. Those sites s that can be updated in parallel are determined as follows. Let s^* be any set of sites and $Q_{s^*} = \{s | s \in Q_r, r \in s^*\}$ be all neighbors of s^*. A *color* is a set s^* such that $s^* \notin Q_{s^*}$. The *lattice chromaticity*[31, 6] is the minimum number of sets s^* that fully cover the Mrf lattice S. Each site $s \in s^*$ can be computed simultaneously. Figure 6d shows the lattice colors for the 8-connected neighborhood; the degree of parallelism is $\mathcal{N}/4$.

The implementation of the Gibbs sampler[5] as a parallel computation is referred to as the *parallel Gibbs sampler* (PGS). The parallel implementation computes

$$P(X_{s^*} = \omega^{s^*} | X_r = \omega, r \in Q_{s^*}) = \frac{1}{Z_{s^*}} e^{-\frac{1}{T} \sum_{C \in \mathcal{C}_{s^*}} U_C(\omega^{s^*})} \tag{3}$$

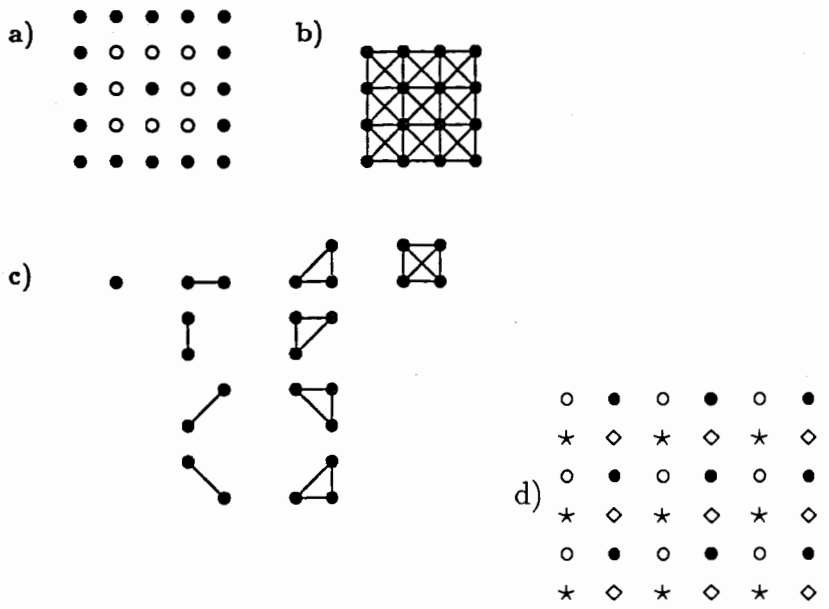


Figure 6: An 8-connected ($c = 2$ or next-nearest-neighbor) Mrf graph with cliques and lattice chromaticity: a) The Mrf lattice (open and filled circles) and the next-nearest-neighbor sites shown with open circles, b) The corresponding Mrf graph, c) All clique types for the 8-connected neighborhood, and d) the chromaticity.

with

$$Z_{s^*} = \sum_{\omega \in \Lambda} e^{-\frac{1}{T} \sum_{C \in \mathcal{C}_{s^*}} U_C(\omega)}.$$

Clique energies are given by $U_C(\omega)$; the sum in the exponential is over the cliques containing the site s. At each iteration in the relaxation, a new state, ω^{s^*}, is generated that may differ from the old state, ω, at every site $s \in s^*$. This new state is generated by sampling the distribution of Equation 3. Sampling this distribution at a single site s requires computation of the exponential and $U_C(\omega^s)$ for all $x_s \in \Lambda$.

The essence of the PGS algorithm is that, besides the parallelism over the lattice sites, computation of Equation 3 is parallel *over the site state space,* Λ. Note that computation of Z_s is inherently serial over Λ. However, for hypercube connected parallel computers, such as the Connection Machine, Z_s is computed as $O(\ln l)$ (where l is the size of the site state space). Also, in practice, computation of $U_C(\omega^s)$ and the exponential is most time consuming; whereas the sum in Z_s over Λ is trivial. Obviously, for large site state spaces the improvement in computation can be significant.

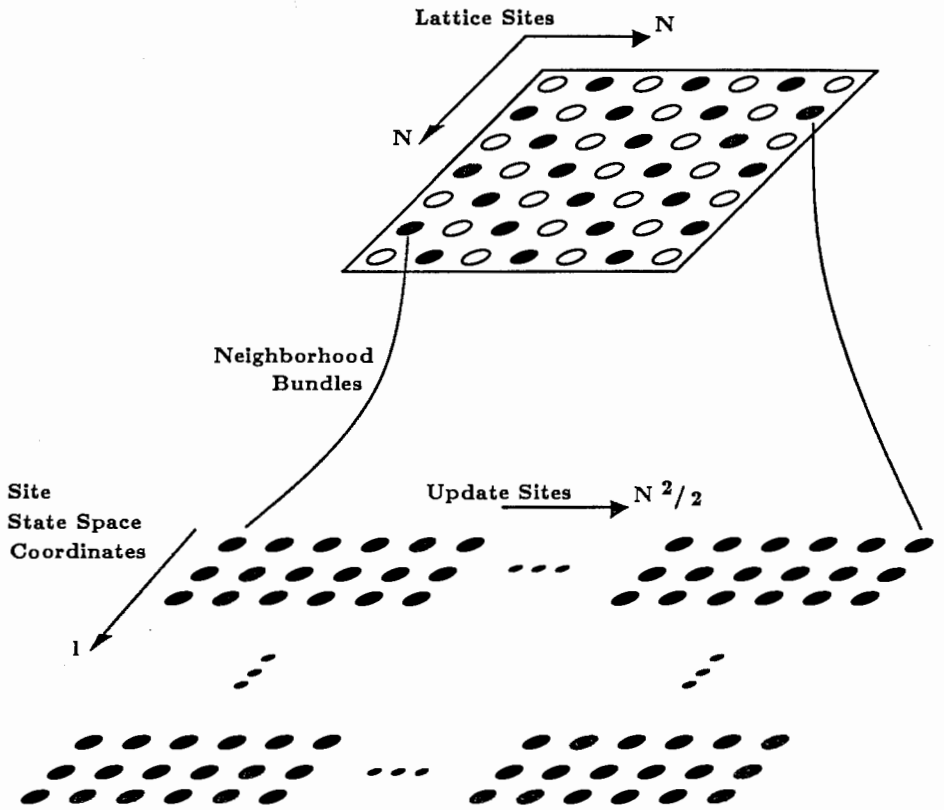


Figure 7: An illustration of the CM configurations for the computation of the PGS algorithm on a 4-connected neighborhood. The top grid is the CM layout for the *image configuration*. Each circle, filled or empty, represents a CM processor associated with an image pixel. The bottom grid is the CM layout for the *PGS configuration*; it is two dimensional: one dimension with a coordinate for each pixel of a given color, and anther dimension with a coordinates for each state in the lattice state space. For the PGS configuration, all processors along the state dimension compute the Mrf energy simultaneously *with the same neighborhood configuration*. The neighborhood configuration at a site s is represented by a "bundle" that is passed between the image configuration and the corresponding site in the PGS configuration.

The PGS is implemented by allocating one CM processor for each pair of site, $s \in s^*$, and lattice state, $\omega_s \in \Lambda$. This layout is displayed in Figure 7. For the PGS layout, the CM configuration is two dimensional with one *site dimension* and another *state dimension.* Each processor along the state dimension in this PGS layout contains the Mrf fields, X_s and Y_s, and the neighborhood fields $\{X_r | r \in Q_s\}$. These fields are required to compute Equation 3. The performance of the PGS algorithm is described elsewhere[32].

2.3 MRF Energy for Discontinuity Detection

The previous sections described the Mrf external field, $Y = \{G, E\}$, the Mrf fields representing the surface, $X = \{F, L\}$, and the stochastic relaxation algorithm. In this section the Mrf posterior energy that relates these fields is presented. This posterior energy embodies our assumptions regarding smoothness of surface properties and discontinuities as well as the assumptions underlying the integration of intensity edge information which facilitates the detection of discontinuities.

The Mrf posterior energy at site s is related to the clique energies of Equation 3 by $U_s(X|Y) = \sum_{C \in \mathcal{C}_s} U_C(\omega)$. For our discontinuity detection scheme this posterior energy is

$$U_s(X|Y) = \sum_{s' \in Q_s} (f_s - f_{s'})^2 \delta_{l_s l_{s'}} (\lambda + \lambda' l_s) + (f_s - g_s)^2 \gamma_s (\alpha + \alpha' l_s) + l_s(\beta + \beta' e_s) \qquad (4)$$

The neighborhood Q_s for this formulation is 8-connected and the only cliques with a non-zero potential are the pair cliques shown in Figure 6. We show the equation in its most general form in which smoothing of F occurs *both* in regions ($l_s = 0$) and along contours ($l_s = 1$) with potentially different couplings of λ and $(\lambda + \lambda')$ respectively. This first term ensures that F is smooth ($f_s \sim f_{s'}$) between neighboring sites where $\delta_{l_s l_{s'}} = 1$. When $\delta_{l_s l_{s'}} = 0$ the smoothing between sites s and s' is broken and this first term does not contribute to the posterior energy.

The second term in this equation is the coupling between the feature process, F, and the sparse and noisy input data, G.. The coupling factors, α and α', are related to the noise in g. For noiseless data, $\alpha \rightarrow \infty$ thereby ensuring $f_s = g_s$. Otherwise, when $\alpha = 0$ no input data coupling occurs and f is smoothed by the term involving $(f_s - f_{s'})^2$. Once again

different couplings between F and G in regions and along contours can be imposed with the term $(\alpha + \alpha' l_s)$. The precise relation between α and the noise depends on the noise model assumed. As this noise process is unknown, values of α and α' are empirically determined.

The last term in Equation 4 implements the integration scheme between surface property discontinuities and intensity edges. The goal for this term is to encourage discontinuity formation where intensity edges exist ($e_s = 1$). The parameters β and β' are chosen to facilitate discontinuity formation when $e_s = 1$ and to forbid discontinuity at sites where $e_s = 0$. Conseqently, $0 < (\beta + \beta') << \beta$. The reduced penalty at an intensity edge produces an increased probability for discontinuity formation. The penalty, $(\beta + \beta')$, to form a discontinuity at an intensity edge should be comparable to the penalty for smoothing ($\sim \lambda(f_s - f_{s'})^2$) when the difference $(f_s - f_{s'})$ is larger than the noise. In this way, surface property changes larger than the noise should form a discontinuity; whereas, smaller changes should be smoothed.

3 Results for Discontinuity Detection

In this section results obtained with the Mrf formulation described in the previous section are presented for a variety of images. As noted previously, discontinuities are detected in a two-phase process: first smoothing to eliminate the displacement errors and then discontinuity detection. The Mrf formulation embodies both phases. Before illustrating the discontinuity results, we describe the "free-field" behavior of the Mrf and then describe briefly the Mrf initialization and smoothing procedures.

Free Field *Free Field* refers to the Mrf response when no input is provided and it measures the Mrf's compliance with the prior assumptions embodied in the posterior energy of Equation 4. The free field is produced by setting $g_s = 0$ and $e_s = 1$ for all $s \in S$. The PGS algorithm is then used to generate the invariant distribution of the Mrf; the ergodicity of the stochastic relaxation simplifies the computation of MAP or MPM states[6].

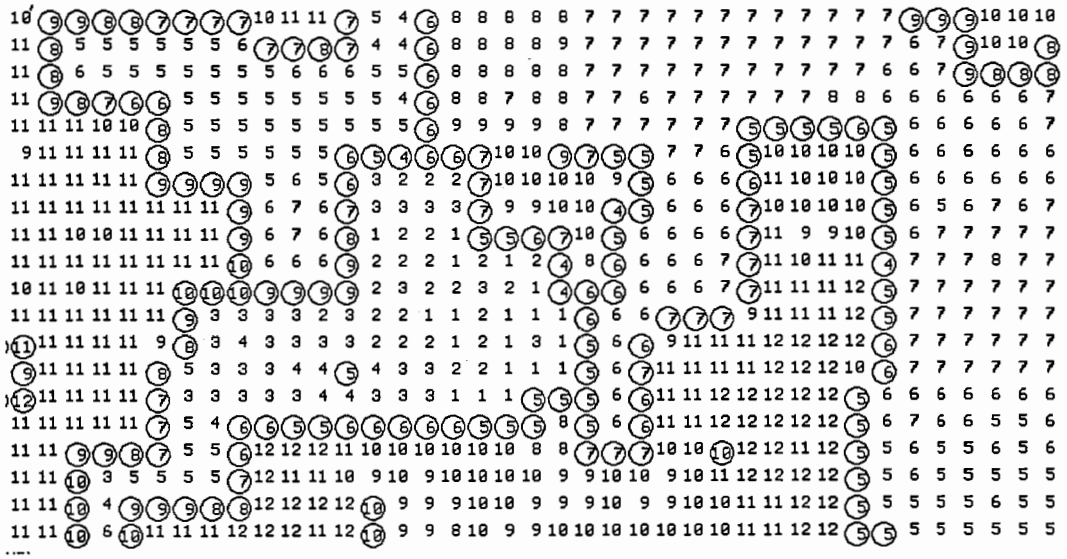


Figure 8: Closeup of the Mrf free field for $(\beta + \beta') = 5.0$, $\lambda = 1.0$ and $(\lambda + \lambda') = 0.1$ in Equation 5.

For the free field, the Mrf's posterior energy becomes

$$U_s(X|Y) = \sum_{s' \in Q_s} (f_s - f_{s'})^2 \delta_{l_s l_{s'}} (\lambda + \lambda' l_s) + l_s(\beta + \beta' e_s) \tag{5}$$

The PGS algorithm used a linear annealing schedule for the temperature that decreased from $T = 25$ to $T = 1$ over 100 iterations. After 100 iterations the temperature was fixed at $T = 1$ and statistics were gathered for the feature process, F, and the discontinuity processes, L. The discontinuity process was binary, $l_L = 2$ and the feature process used $l_F = 16$; the total sites state space size was $l_X = 32$. The PGS algorithm operates across the entire state space simultaneously rather when alternating between the feature and discontinuity process as is common is most relaxation algorithms based upon a dual lattice discontinuity process. The initial Mrf state for F and L was random and the neighborhood was 8-connected. The lattice size was 256 x 256 pixels.

Figure 8 illustrates a free field state produced by this energy. The computed discontinuities, L, are relatively smooth and continuous. These two properties extend over several pixels. Within the regions and along the contours, the surface, F, is smooth. The choice of

parameters for this free field Mrf is empirical. The extent of the smooth regions and the number and length of discontinuities can be varied to a limited extent by adjusting $(\beta + \beta')$[32]. Also, it is generally the case that $(\lambda + \lambda') \sim 0$ in the first term of Equation 5. This condition ensures that when $l_s = 1$ the contribution from this term is nearly zero but that some coupling between the feature process will smooth along the contour as well. Results such as those in Figure 8 guide the selection of parameters for the discontinuity detection.

MRF Initialization Successful detection of discontinuities rests, in part, upon the initialization of the Mrf. With a thoughtful initialization, the convergence time for the Mrf to reach a steady state is reduced and Mrf algorithmic concerns, such as annealing rates, and statistics gathering, are less critical. This improvement arises when the initial state is close to the desired result. With integration, the intensity edges serve as the initial postulate for the surface property discontinuities. With this postulate, the surface property data can be smoothed to eliminate displacement errors.

Figure 9 portrays the raw input data from a feature-based stereo algorithm[2]. The pictures labeled "Sparse Depths" and "Depths Exist" are the outputs from the stereo algorithm. "Depths Exist" is a binary field corresponding to γ_s (in Equation 4) whereas the "Sparse Depths" corresponds to g_s. The picture labeled "Intensity Edges" of Figure 9 are derived from the intensity image with an edge detector, typically Canny's[33]. Before serving as the external field E, the intensity edges are made 4-connected, as required by the 8-connected neighborhood and the form of Equation 4, to fully decouple neighboring depths. These types of inputs comprise the external field Y and are common for the different images presented subsequently.

Following these inputs, the Mrf field, $X = \{F, L\}$ can be initialized. Theoretically, X should be randomized (or the initial temperature should be high enough to randomize X) thereby guaranteeing convergence to the global minimum of $U(X|Y)$. However, by selecting an initial state that is close to the desired result, the computational requirements are reduced. We choose the initial state to exhibit the smoothness and continuity already embodied in Equation 4. The discontinuity process, L, is initialized to the 4-connected intensity edges E. With this initialization, L is both smooth and continuous. Since the input surface properties

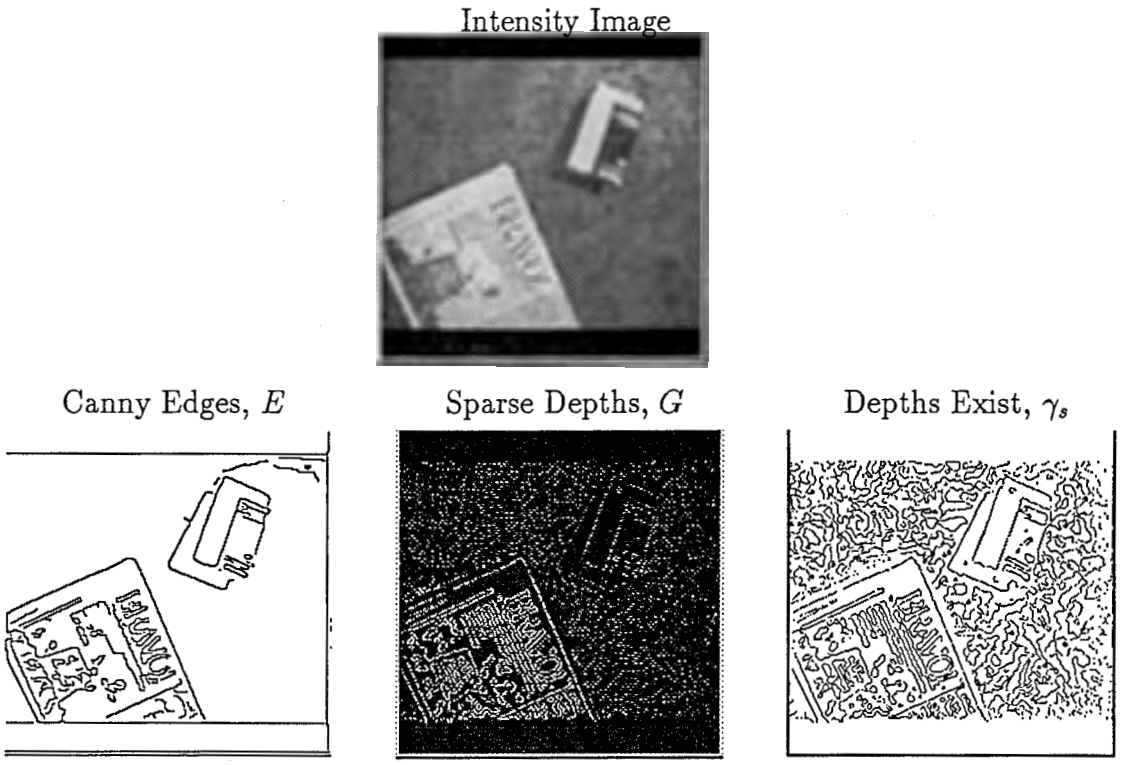


Figure 9: The "Intensity Image" is the left image of the original stereo pair. The other three pictures are the raw inputs for discontinuity detection and are labeled with the Mrf field that they represent.

are sparse, the feature process F must be filled for initialization.

Filling the regions in F is accomplished by growing about the pixels where depth data exists. This is "brush fire filling." However, since the initial discontinuity field is known, this filling is performed while inhibiting growth across a discontinuity. The discontinuities constrain the filling and thereby ensure that a discontinuity is not smoothed away. The first column in figure 10 illustrates the results of this initialization. The top picture in this column is the filled depth map; whiter pixels are closer to the camera. The two pictures below are close-ups of two small regions within the 256 x 256 image. The squares show the sparse depths; the circles are the intensity edges. Note that this filling procedure correctly restrains the surface except when a displacement error occurs. As the middle picture (at pixel (90 85)) illustrates, displacement errors still persist.

3.1 Depth and Motion Discontinuities

The First Phase: Stochastic Smoothing Recall the goal of this first phase: eliminate the displacement errors. The indicated discontinuity at $i_{s'}$ must be shifted to the actual discontinuity at a_s. Of course the actual discontinuities, A, are unknown; but, since we assume $A \subset E$, we can shift the indicated discontinuities to e_s. This technique, as detailed previously, is one of the primary advantages from integrating intensity edges with surfaced properties and its use is mandated by the pervasive displacement errors. The shift is accomplished within the Mrf framework by smoothing F while temporarily fixing $L = E$. The PGS algorithm performs the smoothing stochastically; fixing $L = E$ is accomplished by annealing the parameters β and β of Equation 4. These parameters determine the difference in surface property required to turn on a discontinuity. If $(\beta + \beta') << 0$ and $\beta >> 0$, then $L = E$ is fixed as the stochastic relaxation algorithm proceeds. Only F varies and, when α is small, the surface properties are smoothed.

The second column in Figure 10 illustrates the effect of stochastic smoothing for the data of Figure 9. The middle picture which was previously dominated by a displacement error (compare with the first column filling results) is no longer plagued by this error. Instead the discontinuity has been correctly shifted. The discontinuity process prevented smoothing of the surface properties across intensity edges; the indicated discontinuities now appear

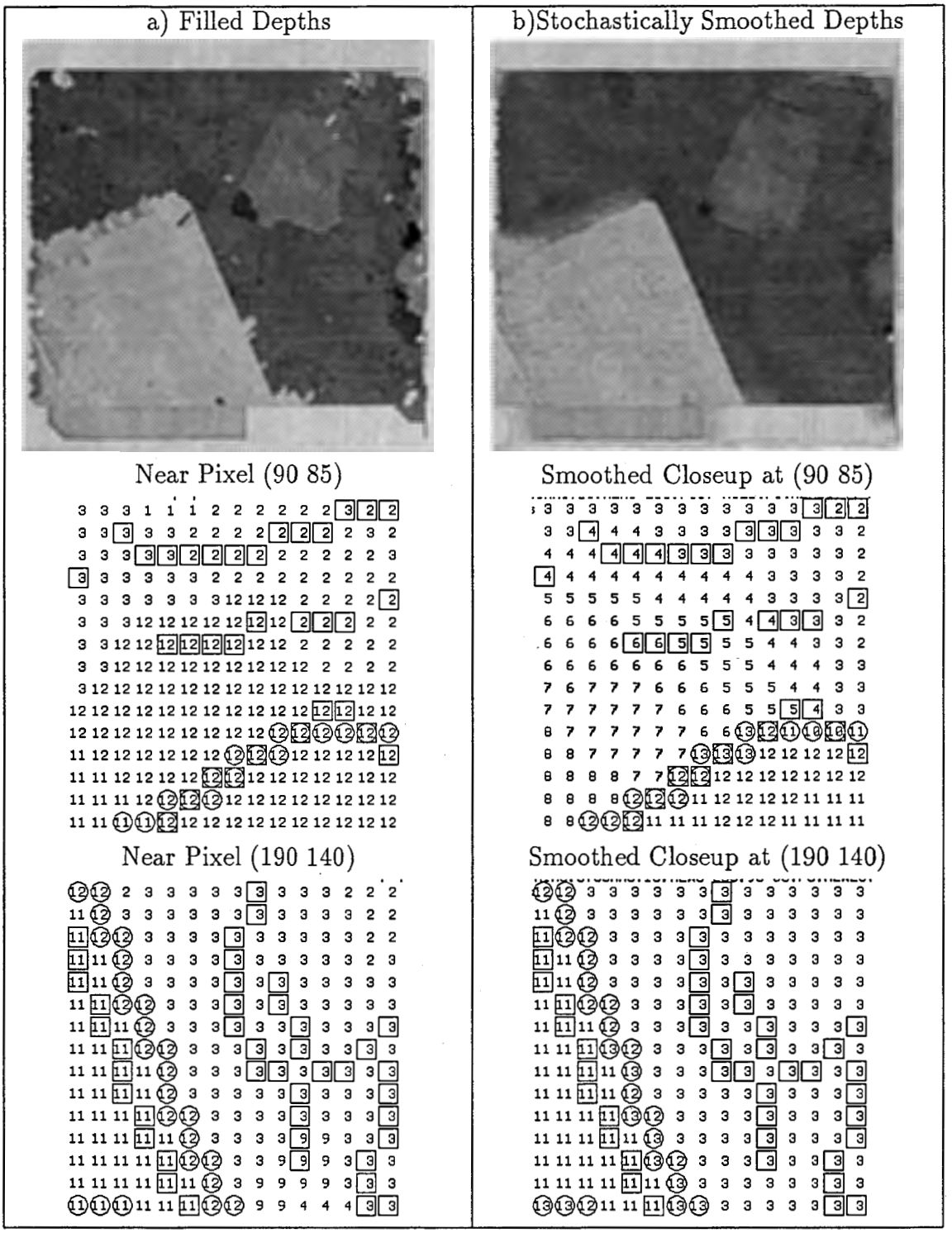
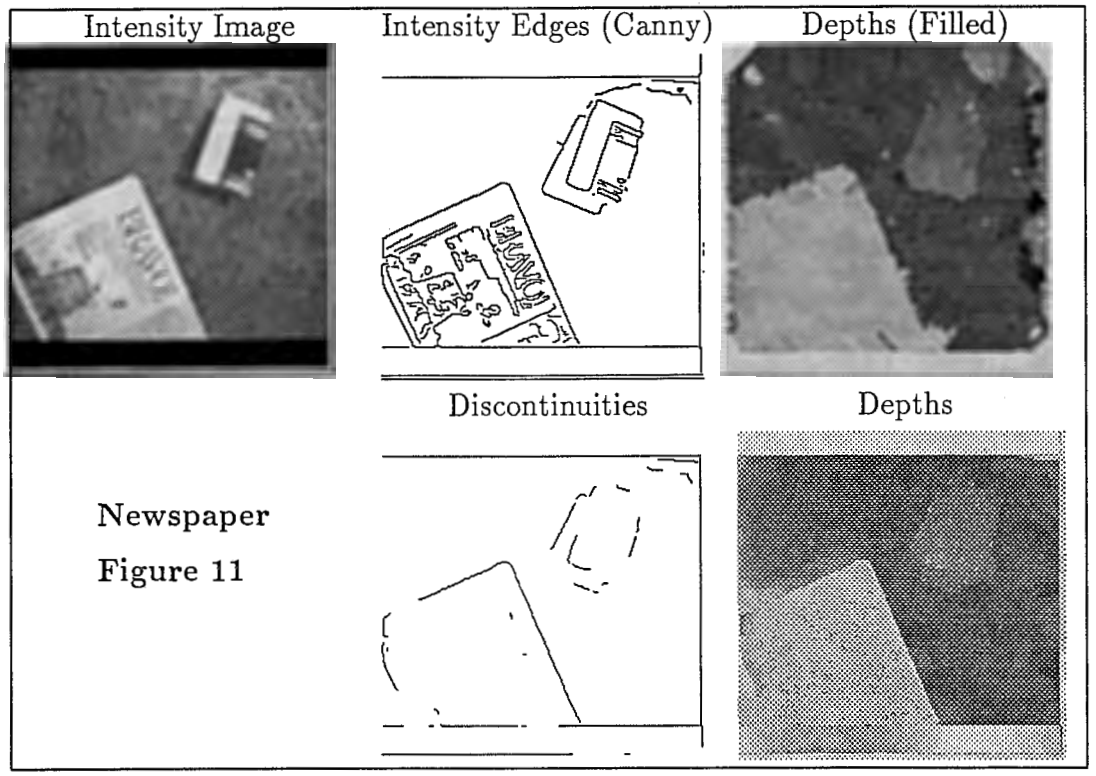


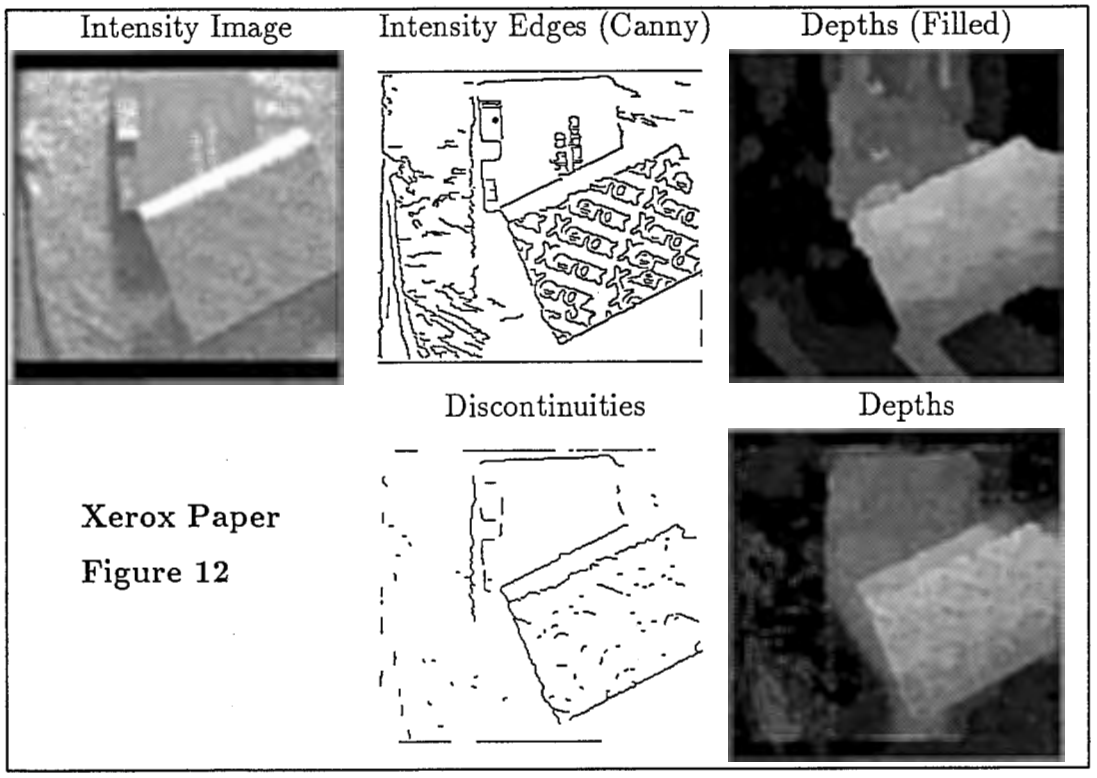
Figure 10: The depths in the "Newspaper" scene are filled as shown in the first column (a), and then stochastically smoothed as the second column illustrates (b). Close-ups of the Mrf fields, X and Y, are also shown.



coincident with the intensity edges.

The Second Phase: Discontinuity Detection Results Figure 11 shows the object boundary identified by the Mrf when intensity edges are coupled with the depth data from stereo. The format of this figure is used throughout this chapter and it warrants a brief description. The upper row of Figure 11 contains the Mrf inputs that comprise the external field. Depending on the early vision module, the intensity image is one of the stereo pair or the motion sequence. The intensity edges, E, and the initial state for L are derived from Canny's edge detector. The "filled" picture in the upper row is the initial state of F; its derivation was described and illustrated in detail in the previous sections. The lower row in Figure 11 is the output from the Mrf process. The "depths" picture is the field F and the MPM estimate for the discontinuity process is displayed in the "discontinuities" picture.

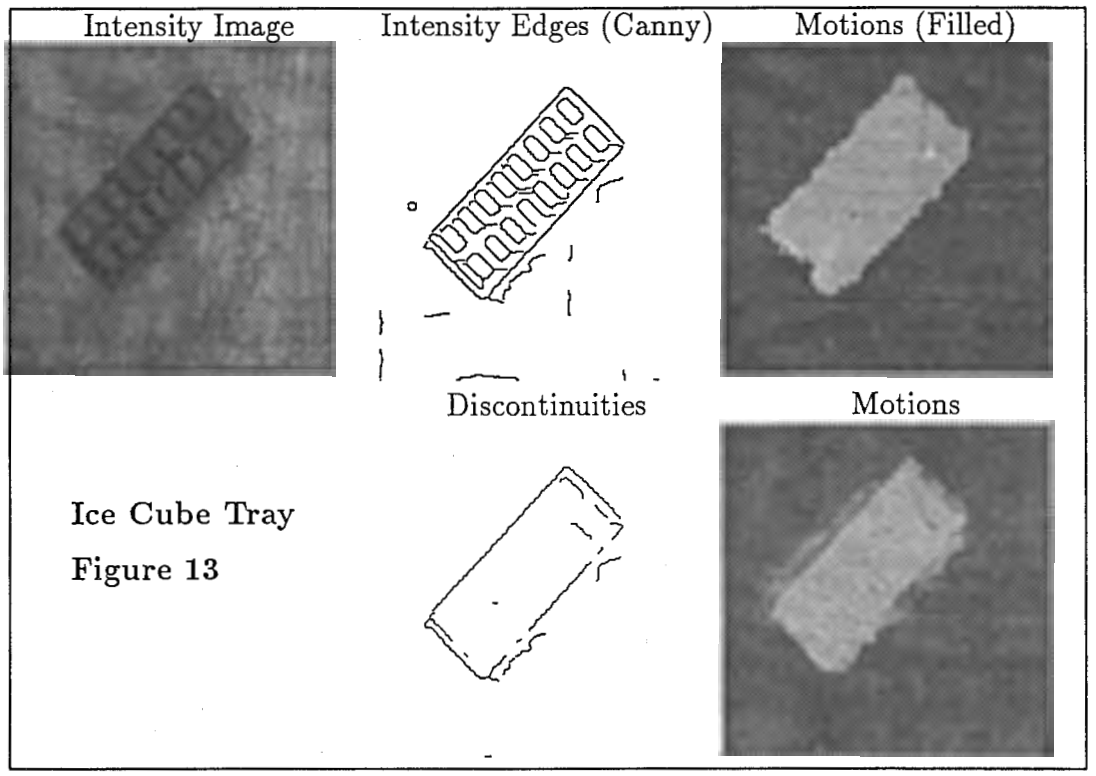
In Figure 11 discontinuities are well localized, smooth and continuous. A significant displacement error along the top edge of the stack of newspapers is smoothed away and its depth discontinuity is located properly. The significantly small object in the upper right is



partially extracted; texture, unrelated to depth discontinuities, is eliminated.

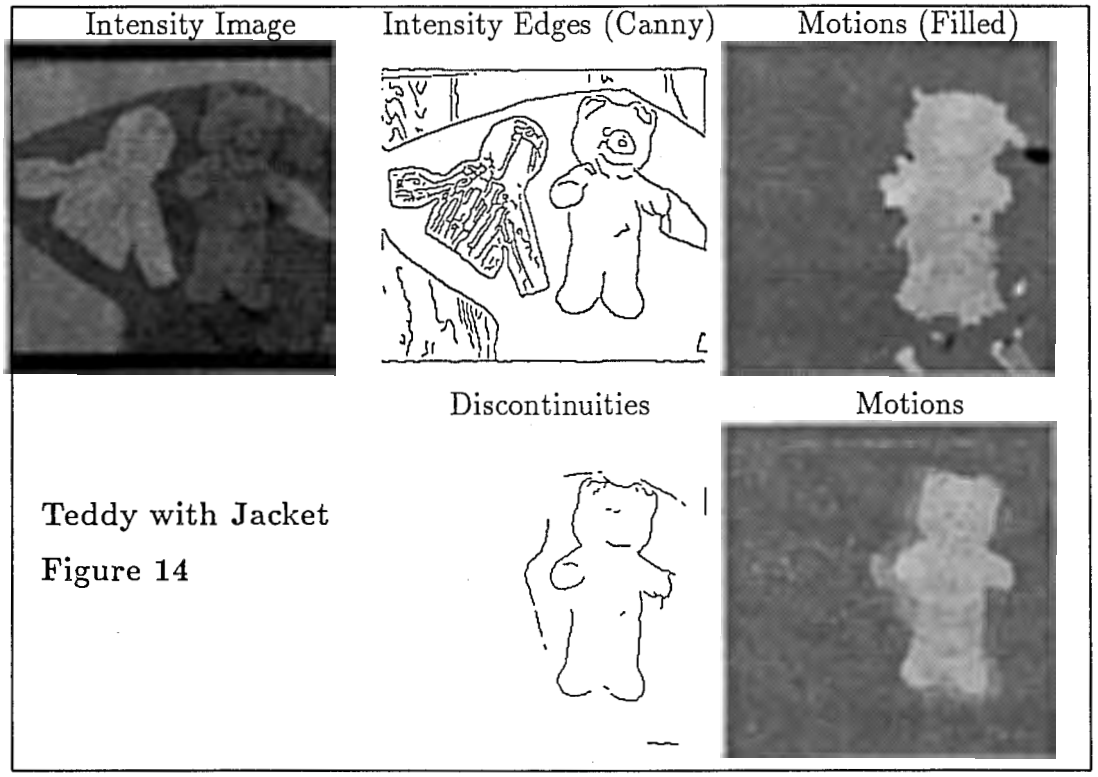
The example of Figure 12 is more difficult for this integration scheme than the previous figure. Figure 12 contains several objects each with different magnitudes of depth discontinuity. One object is sloped significantly and, because the sloped surface also contains significant texture, could yield multiple, misplaced boundaries. However, in spite of this, the integration scheme still yields a reasonable map of discontinuities. Notice the displacement errors near the boundaries in the "depths" picture of Figure 12. In particular, the left side, for the left camera in a stereo pair, cannot match the occluded region. Consequently, a displacement problem arises and the smoothing of the Mrf helps constrain the discontinuities to the intensity edges.

The next set of images deal with the optical flow data[13]. The optical flow is a vector field in the 2D image plane. The vector indicates the displacement between corresponding pixels in a time sequence of images. As an external field, G, for the Mrf, the magnitude of the optical flow is used. Consequently, only discontinuities in the magnitude of the optical flow are identified. The random field F remains a scalar field.



Ice Cube Tray

Figure 13



Teddy with Jacket

Figure 14

Figures 13 and 14 show examples of motion discontinuity detection. Comparison between the filled motion picture and the motion picture in these two figures once again illustrates the displacement errors and the smoothing required to eliminate this error. In practice, the motion algorithm produces a noisier result compared to the stereo algorithm. However in these cases, the motion discontinuities are accurately computed.

3.2 Without Intensity Edges

Figure 15 compares discontinuity detection with and without intensity edges for several images. As this figure illustrates, when intensity edges are not integrated with surface property data, the detection of discontinuities deteriorates. Without intensity edges, E is zero for all sites $s \in S$ and the penalty to form a discontinuity becomes β throughout the image. In addition, without intensity edges, the initialization process changes. Brush fire filling is used to make the sparse input data, g_s, dense; but, filling while respecting the intensity edges and deterministic smoothing can no longer be used. Because of displacement errors, indicated discontinuities are found; not actual discontinuities. All four images of Figure 15 are plagued by this type of error. Also, several small discontinuities are missed compared to results with intensity edges.

In general, the discontinuities identified without intensity edges are not as smooth as when integration is performed. Increasing the clique size for the discontinuity process by one or two pixels does not significantly improve the discontinuity smoothness. In some situations, the discontinuities produced without integration are continuous or closed relative to discontinuities with integration. This additional continuity sometimes produced without integration is of little consequence compared to the accuracy achieved with integration.

4 Summary

We have detected discontinuities in depth and in the magnitude of optical flow for a variety of natural images by combining intensity edges and surface property data computed with early vision algorithms. The detected discontinuities are consistently smoother and better localized relative to the discontinuities detected without the benefit of intensity edges.

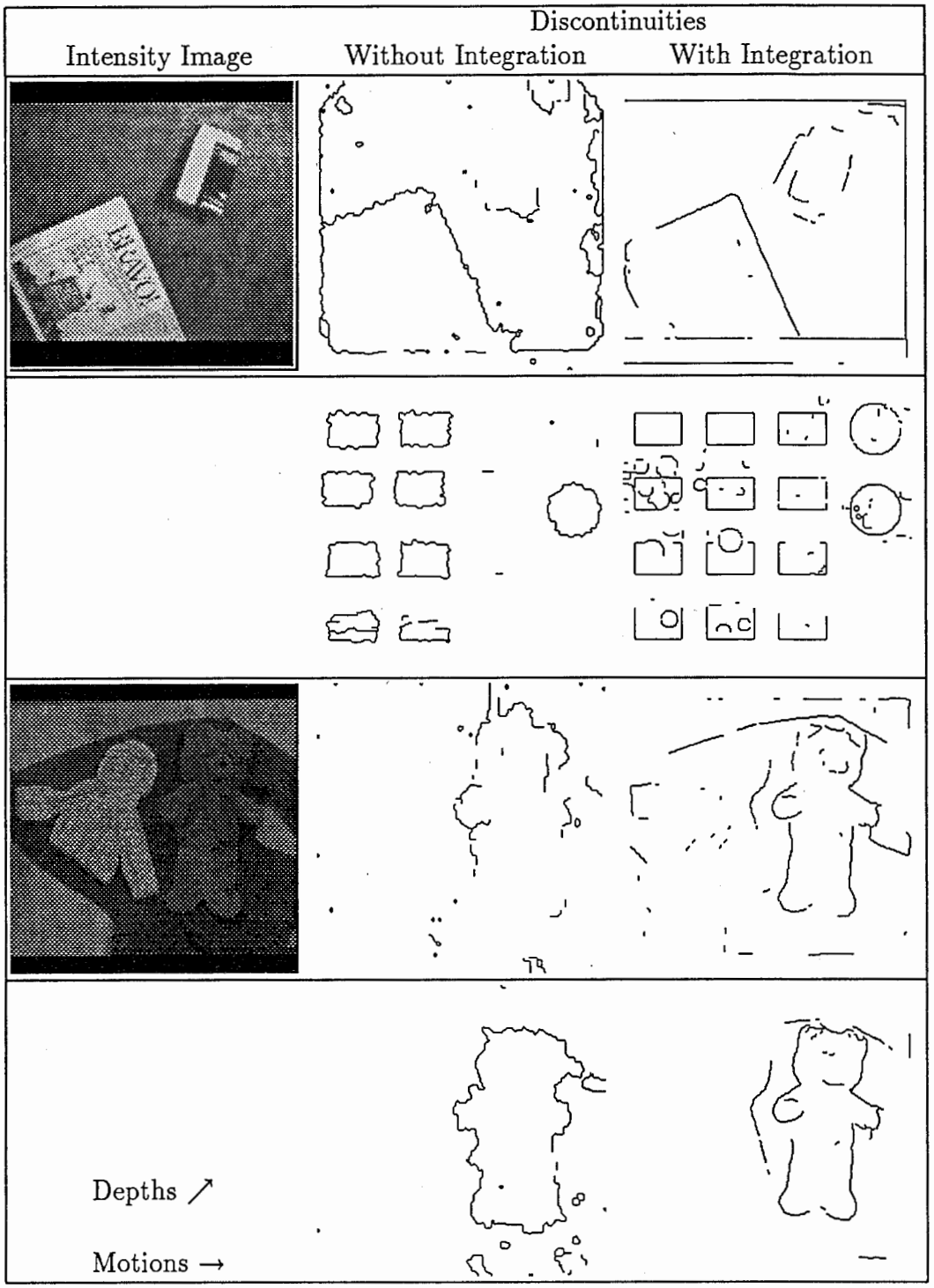


Figure 15: A comparison between discontinuity detection with and without integration of intensity edges. The second row is a synthetic depth image that was designed to test common problems in discontinuity detection. Namely: displaced data, missing intensity edges, steeply sloped surfaces and circular boundaries. The bottom two rows contain the discontinuities in depth and motion for the identical image shown.

There are two major differences between this work and prior work[18] on integrating intensity edges with surface property data. First, here the discontinuity detection is performed explicitly in light of the capabilities of recognition algorithms. This suggests that, since most model-based recognition algorithms recognize objects based on shape and handle contour occlusion, a discontinuity detector that potentially sacrifices some contour pixels in order to produce smooth and well localized contours is preferred. Second, over the past years it has become apparent that the pervasive errors in surface properties computed by early vision algorithms are not Gaussian distributed errors at every pixel but rather displacement errors at discontinuities. Gaussian distributions as models of error may have been adopted in Mrf techniques for surface reconstruction not for their relevance to the noise in real surface property data but instead for their solvability. These two factors reshape the surface reconstruction problem with important simplifications.

Herein we have described the conditions underwhich the surface reconstruction can be simplified. The primary assumption is that surface property changes originate intensity variations. Since in our formulation discontinuities must be a subset of intensity edges, if an intensity edge is not produced at such a change in surface property then the discontinuity will not be detected. Aside from systematic cases like texture, this situation can arise due to imaging errors, such as underexposure or overexposure, or from edge detector shortcomings. However, recognition algorithms operate even when image contours are missing; consequently, the affect of missed contours due to edge detector failings is minimized by the capabilities of recognition algorithms.

The simplification and results presented are as follows. First, since discontinuities are a subset of intensity edges, the detected discontinuities are as smooth and well localized as the intensity edges. Without the need to impose smoothness on the discontinuities, the entire notion of "line process clique configurations"[5, 27] can be discarded. Second, the notion of the dual lattice is discarded in favor of a single, binary discontinuity process. Third, displacement errors are eliminated based on a Mrf formulation that first smoothes then detects discontinuities. This approach is mandated by the displacement errors and produces good results in the Mrf formulation. Other non-Mrf formulations may be possible[29, 35] based on these simplifications. Besides these results, we briefly have also presented the

parallel Gibbs sampler algorithm. This stochastic relaxation algorithm improves the solution of optimization problems when the site state space is large.

References

[1] Anselm Spoerri and Shimon Ullman. The early detection of motion boundaries. In *Proceedings of the International Conference on Computer Vision*, pages 209–218, London, England, June 1987. IEEE, Washington, DC.

[2] Walter Gillett. Issues in parallel stereo matching. Master's thesis, Massachusetts Institute of Technology, 1988.

[3] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems.* W.H.Winston, Washington, D.C., 1977.

[4] W. Eric L. Grimson. *From Images to Surfaces.* MIT Press, Cambridge, Mass., 1981.

[5] Stuart Geman and Don Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, 1984.

[6] Jose L. Marroquin. *Probabilistic Solution of Inverse Problems.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1985.

[7] Andrew Blake and Andrew Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, Mass, 1987.

[8] Demetri Terzopoulos. The role of constraints and discontinuities in visible-surface reconstruction. In *Proceedings IJCAI*, pages 1073–1077, August 1983.

[9] John Y. Aloimonos. Unification and integration of visual modules: An extension of the Marr paradigm. In *Proceedings Image Understanding Workshop*, pages 507–551. Morgan Kaufmann, San Mateo, CA, Palo Alto, CA, May 1989.

[10] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman and Company, San Francisco, 1982.

[11] Berthold K. P. Horn. *Robot Vision.* MIT Press, Cambridge, Mass., 1986.

[12] Tomaso Poggio and Vincent Torre. Ill-posed problems and regularization analysis in early vision. A.I. Memo No. 773, C.B.I.P. Paper No. 001, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1984.

[13] Heinrich H. Bülthoff, James J. Little, and Tomaso Poggio. A parallel algorithm for real-time optical flow. *Nature*, 337:549 – 553, February 1989.

[14] David Lee. Coping with discontinuities in computer vision: Their detection, classification, and measurement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(4):321–344, April 1990.

[15] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. R. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, New York, 1978.

[16] Tomaso Poggio. Integrating vision modules with coupled MRF's. Working Paper No. 285, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1985.

[17] Paul B. Chou and Christopher M. Brown. Multi-modal segmentation using Markov random fields. In *Proceedings Image Understanding Workshop*, pages 663–670, Los Angeles, CA, February 1987. Morgan Kaufmann, San Mateo, CA.

[18] Edward B. Gamble and Tomaso Poggio. Visual integration and detection of discontinuities: The key role of intensity edges. A.I. Memo No. 970, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, October 1987.

[19] R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1):67–108, March 1986.

[20] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, 1985.

[21] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[22] W. Eric L. Grimson and Tomas Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):469–482, July 1987.

[23] Daniel P. Huttenlocher and Shimon Ullman. Object recognition using alignment. In *Proceedings of the International Conference on Computer Vision*, pages 102–111, London, England, June 1987. IEEE, Washington, DC.

[24] W.E.L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. In *Proceedings of the International Conference on Computer Vision.* IEEE, Washington, DC, 1988.

[25] Shimon Ullman and Amnon Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connected network. A.I. Memo No. 1061, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, July 1988.

[26] J. L. Marroquin and T. Poggio. A markovian random field of piecewise straight lines. preprint, 1988.

[27] Bernard W. Silverman, Christopher Jennison, Julian Stander, and Timothy C. Brown. The specification of edge penalties for regular and irregular image pixels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):1017–1024, October 1990.

[28] J. Hutchinson, C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computer Magazine*, 21:52–64, 1988.

[29] Davi Geiger. *Visual Models with Statistical Field Theory.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[30] B. Gidas. A renormalization group approach to image processing problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11:164–180, 1989.

[31] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc.*, B34:75–83, 1972.

[32] E. B. Gamble Jr. *Integration of Early Vision Cues for Recognition.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1990.

[33] John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[34] Richard C. Dubes and Anil K. Jain. Random field models in image analysis. *Applied Statistics*, to be published 1989.

[35] E. B. Gamble Jr. Augmented binomial convolution for discontinuity detection. in preperation.