Line Labeling Using Markov Random Fields

Terry Regier*

TR-91-059

November 1991

Abstract

The task of obtaining a line labeling from a greyscale image of trihedral objects presents difficulties not found in the classical line labeling problem. As originally formulated, the line labeling problem assumed that each junction was correctly pre-classified as being of a particular junction type (e.g. T, Y, arrow); the success of the algorithms proposed have depended critically upon getting this initial junction classification correct. In real images, however, junctions of different types may actually look quite similar, and this pre-classification is often difficult to achieve. This issue is addressed by recasting the line labeling problem in terms of a coupled probabilistic system which labels both lines and junctions. This results in a robust system, in which prior knowledge of acceptable configurations can serve to overcome the problem of misleading or ambiguous evidence.

*This work was supported by ONR contract N00014-85-K-0692 and ONR grant N00014-89-J-1251 to Donald Glaser of UC Berkeley. The author gratefully acknowledges support from him, and from Jerome Feldman of the International Computer Science Institute in Berkeley. Jitendra Malik of UC Berkeley suggested the application of MRFs to the line labeling problem.

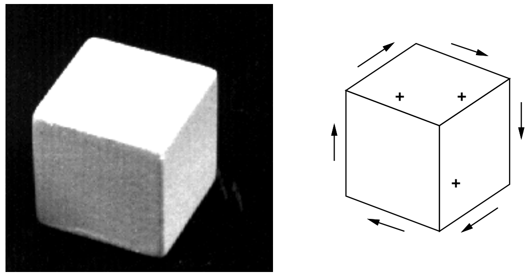
(a)  (b)

Figure 1: Line Labeling from an Image

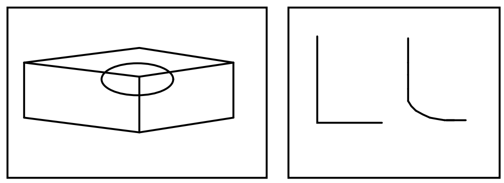
(a)  (b)

Figure 2: Perceptual Similarity of Junction Types

1 Introduction

Given a greyscale image of solid, opaque polyhedra with exactly three planes touching at every vertex, we wish to obtain a line labeling for the image. This is illustrated in Figure 1, where (a) is the input image, and (b) is the line labeling produced for that image. The labels used here are based on the well-known work of [Huffman, 1971; Clowes, 1971].

In this task, there are a number of difficulties which are not present in the original formulation of the line labeling problem, or in subsequent work [Waltz, 1975; Malik, 1985]. It has generally been assumed that input will be in the form of an idealized line drawing, with each junction correctly pre-classified into one of several types (e.g. T, Y, arrow, etc). The algorithms proposed have been critically dependent on the correctness of this *a priori* junction classification.

In an actual image of a set of objects, however, junctions of different types may be perceptually quite similar, and thus difficult to classify on the basis of perceptual evidence alone. Figure 2 presents two examples of this. In (a), the circled Y junction is flat enough to be quite similar to a T junction. If the junction were misclassified this way, the classical line labeling schemes would be worthless. In (b), we see two junction types taken from [Malik, 1985], an extension of early line labeling work to cover the case of piecewise smooth curved objects. The two junction types are perceptually very similar, but place different constraints on the labels of the incoming edges. As Malik points out, it is probably asking too much of a separate front-end process to be able to consistently distinguish these.

Thus, quite apart from issues of edge detection and junction detection, this problem of the perceptual similarity of junction types must be addressed by any work which seeks to provide an account of line labeling from images.

2 A Coupled System

The approach taken by this work is to reformulate the line labeling problem in terms of a coupled probabilistic system. Under this scheme, we do *not* assume correctly pre-classified junctions, as earlier work has. Instead, the issue of perceptual similarity of junction types is addressed by labeling both lines *and* junctions. Once lines and junctions are detected, external evidence for particular junction and line labels is extracted from the image, and this, together with prior probabilities for given configurations of junction and line labels, gives us a posterior distribution over all possible labelings. The idea is to have the system arrive at the labeling with the maximum *a posteriori* probability.

This provides a solution to the problem of perceptual similarity of junction types, since any junction in the image which is not clearly of a single junction type will provide weak evidence for all possible labels. For example, in Figure 2(a), the circled junction will provide weak evidence for both the "T" label and the "Y" label. The "Y" label will eventually be chosen since it is more consistent with the rest of the image, i.e. because there exists a resulting overall labeling with higher *a posteriori* probability than any overall labeling in which that junction is labeled as a "T". Note that since the *a posteriori* distribution over labelings is derived through Bayes' rule from the *a priori* distribution, we are actually letting the prior probabilities of particular configurations of line and junction labels help us determine the label for junctions like this, rather than relying on perceptual evidence alone.

Line labels are determined in the same fashion.

3 Markov Random Fields

Markov random fields (MRFs) are used to implement the above ideas. This formalism has found several uses in machine vision recently [Cross and Jain, 1983; Geman and Geman, 1984; Chou and Raman, 1987; Cooper, 1989]. Before proceeding to outline the details of the system, we present a brief introduction to MRFs, as used here.

Let S be a set of sites connected through an undirected graph, called the *neighborhood graph*, and let $X = \{X_s, s \in S\}$ be a set of random variables indexed by S. Adjacent vertices in the graph correspond to neighboring MRF elements. We assume, without loss of generality, that there exists a state space (or label space) L common to all the variables, such that the value ω_s of X_s is in L. The term ω denotes an assignment of some value from L to each element of X, and is referred to as a *configuration*.

X is a Markov random field if and only if the following two properties hold:

Positivity:

$$P(X = \omega) > 0, \forall \omega \tag{1}$$

Locality:

$$P(X_s = \omega_s | X_r = \omega_r, r \in S, r \neq s) = P(X_s = \omega_s | X_r = \omega_r, r \in N_s) \tag{2}$$

where N_s is the set of sites which are neighbors of s in the graph. The property of locality states that the label at a site is dependent only on the labels at neighboring sites.

An alternate characterization is given by the Hammersley-Clifford theorem: X is a Markov random field if and only if the probability distribution $P(X = \omega)$ is a Gibbs distribution

$$P(\omega) = \frac{e^{-\frac{1}{T}U(\omega)}}{Z} \tag{3}$$

where T is the temperature, U is an energy term, and Z is a normalizing constant. U is obtained by summing over applicable clique potentials $V_c(\omega)$:

$$U(\omega) = \sum_{c \in C} V_c(\omega) \tag{4}$$

It is through assigning potentials to cliques[1] in the neighborhood graph of the MRF that one specifies constraints governing local label configurations. Each time a clique matches a subgraph of the neighborhood graph, its clique potential is added to the sum U.

In the work presented here, we wish to take into account not only the prior probabilities of particular configurations, but also external evidence. The external evidence for a particular label ω_s at a site s after an observation O_s is given by the likelihood $P(O_s|\omega_s)$. We make the assumption that these likelihoods are conditionally independent:

$$P(O|\omega) = \prod_{s \in S} P(O_s|\omega_s) \tag{5}$$

Given this assumption, and starting with Bayes' rule, we get the following derivation of the *a posteriori* probability of a configuration ω given an observation O:

$$\begin{aligned}
P(\omega|O) &= \frac{P(\omega)P(O|\omega)}{P(O)} && (6)\\
&= \frac{\frac{e^{-\frac{1}{T}U(\omega)}}{Z} \times \prod_{s \in S} P(O_s|\omega_s)}{P(O)} && (7)\\
&= \frac{e^{-\frac{1}{T}U(\omega)} \times \prod_{s \in S} P(O_s|\omega_s)}{Z \times P(O)} && (8)\\
&= \frac{e^{-\frac{1}{T}U(\omega)} \times e^{\log(\prod_{s \in S} P(O_s|\omega_s))}}{Z'} && (9)\\
&= \frac{e^{[-\frac{1}{T}U(\omega)]+[\sum_{s \in S} \log P(O_s|\omega_s)]}}{Z'} && (10)\\
&= \frac{e^{-\frac{1}{T}[U(\omega)-T\sum_{s \in S} \log P(O_s|\omega_s)]}}{Z'} && (11)\\
&= \frac{e^{-\frac{1}{T}[\sum_{c \in C} V_c(\omega)-T\sum_{s \in S} \log P(O_s|\omega_s)]}}{Z'} && (12)
\end{aligned}$$

where $Z' = Z \times P(O)$. From this, we see that the posterior Gibbs energy after an observation O is

$$U(\omega|O) = \sum_{c \in C} V_c(\omega) - T \sum_{s \in S} \log P(O_s|\omega_s). \tag{13}$$

[Chou and Raman, 1987] present a deterministic method, the "Highest Confidence First" (HCF) algorithm, for constructing a configuration with a local minimal a posteriori energy measure, giving an estimate to the configuration with maximal a posteriori probability $P(\omega|O)$. The basic intuition behind the algorithm is that initially, all nodes take a special "null" label, and after that, the first sites to be labeled should be those for which the evidence is most decisive in favor of a particular label. HCF tends to provide fast convergence, requiring an average of around one update per node. This is the algorithm used in the work presented here.[2]

[1] Recall that a clique is a completely connected subgraph.

[2] Jitendra Malik of UC Berkeley has pointed out that HCF actually violates the positivity criterion, as there is a configuration under HCF which has $P(\omega|O) = 0$ (namely, the configuration in which all sites have the null label). HCF is thus technically not an MRF implementation. I note this distinction in the interests of accuracy; however, I shall continue to use the term "MRF" in reference to HCF systems, as this is in fact the only violation.

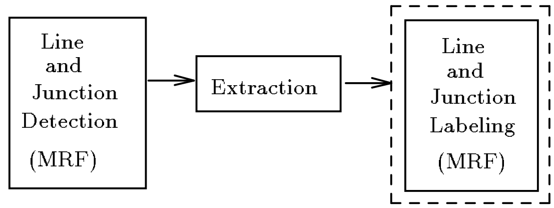


Figure 3: System Architecture

4 Architecture

The architecture of the line and junction labeling system is presented in Figure 3. It consists of three stages: line and junction detection, extraction of lines and junctions from the image, and line and junction labeling. The final stage, highlighted with a dotted outline in the figure, is the coupled system that this report focuses on.

It is important to stress that the first two stages, the detection and extraction modules, in no way consitute a complete solution to the problem of line and junction detection in images. They are included here simply to serve as a general initial indication of the sort of mechanism that might perform these tasks. The final labeling module, on the other hand, is felt to be a reasonably complete solution to the problem of labeling lines and junctions as specified here, *given that lines and junctions have been detected.* Note that the problem solved by this final module is different from the classical line labeling problem, since here, the actual orientations of the lines which meet at a junction are considered to be significant, while in the original task formulation, junctions were simply pre-classified into categories, and the line orientations discarded. As we shall see, retaining the actual orientations allows us to address the issue of perceptual similarity of different junction types.

A more complete solution to the overall problem of line labeling from images would provide robust, error-resistant detection and extraction stages as well; this is a direction for future research.

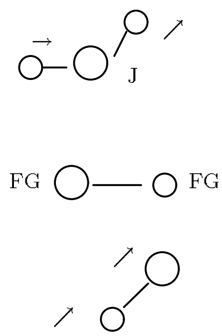
4.1 Detection of Lines and Junctions

The line and junction detection mechanism accepts a greyscale image of trihedral objects on a dark background, and produces an array of labels, such that each pixel has been labeled as one of $\{FG, BG, \Theta_0, \Theta_{15}, ...\Theta_{165}, Junction\}$. Here FG represents foreground, BG represents background, Θ_i represents a line of orientation i degrees, and $Junction$ represents the occurrence of a junction. Note that there are 12 orientations here, one every 15 degrees. This stage is implemented as an MRF.

4.1.1 External Evidence for Lines and Junctions

The first step in the detection of lines and junctions is the computation of $P(O_s|\omega_s)$, the external evidence for each label at each point in the image. To this end, the image is convolved with a set of 5x12 simple oriented filters, one per orientation label. For a given orientation θ and location i, this yields a reasonable estimate to $P(O_i|\theta)$, i.e. the evidence for a line of that orientation at that point.

The evidence for the labels FG and BG at a given position i are determined using the average intensity over a small (3x3) patch of the image, centered on the point in question.



(a) -1.0: Junctions between different orientations

(b) -0.21: Region-growing
(0.21 if not the same label)

(c) -0.2: Oriented line continuity

Figure 4: Cliques for the Detection MRF

$P(O_i|Junction)$ is zero; we start without any external evidence for junctions.

4.1.2 The Line and Junction Detection MRF

The neighborhood graph for the detection MRF is set up so that each node (i.e. each pixel) is connected to all other nodes within a 5x5 square centered on that node. This is done so that the central node together with any subset of the eight immediately surrounding nodes will form a clique, i.e. a completely connected subgraph. All cliques for this MRF take the form of subsets of the central nine nodes.[3]

Representative samples of the cliques used in the system are presented in Figure 4.

In Figure 4, large circles denote the current node being examined, while smaller circles denote neighbors of that node. Junction labels are represented by "J", orientation labels are represented by an arrow in the corresponding direction, and foreground is "FG". Given this, the cliques listed in Figure 4 can be seen to encode the following facts about straight lines and junctions:

- a: Junctions should be called between lines of differing orientation. Note that this clique is strongly weighted, since there is no external evidence for junctions; they must be called into existence on the basis of prior probabilities alone.
- b: If a neighboring node is foreground, the current node probably is as well. However if a neighboring node is background, the current node is probably not foreground. This is essentially the same as the region-growing cliques presented in [Geman and Geman, 1984].
- c: Oriented lines tend to continue at the same orientation.

While considerations of space do not allow an exhaustive listing here of the cliques used by the system, this should serve to convey a sense for the sorts of cliques used.

An example of the final results of this stage is presented in Figure 5, which is a screendump from the MRF simulator built for this work. The object shown is a tall rectangular prism. Different graylevels in the image correspond to different labels, such that the background is black, the foreground white, different orientations take different greylevels in between (but all relatively bright), and junctions are colored dark gray.

[3] Another way of viewing this is that we first determine which subsets of the central nine nodes we want to use as cliques, and then determine a graph structure under which these are in fact cliques.

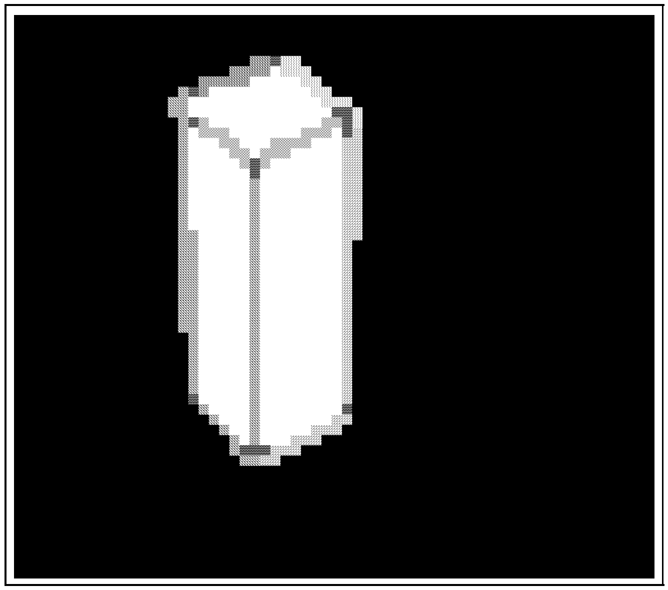


Figure 5: Results of Line and Junction Detection

4.2 Extraction of Lines and Junctions

The line and junction extraction mechanism is responsible for translating the output of the first stage into a format suitable for the final one. It accepts as input the pixel-based array of labels computed by the detection mechanism, and produces a neighborhood graph for the line and junction labeling MRF, in which nodes will represent individual junctions and lines, rather than pixels. It also outputs likelihoods for the various labels of the final MRF. This is done by searching through the pixel-based output of the first stage, looking for individual junctions and lines. It must then

- Create a junction node for each junction found. Since a single junction in the image may give rise to a number of pixels in the image being labeled as junctions (see Figure 5), junction pixels within a certain small distance ϵ of each oter are clustered together, and are interpreted as a single junction. ϵ was 6 pixels in the experiments run for this report.
- Create a line node for each line found.
- Set up the neighborhood graph so that each junction node is connected to the line nodes that correspond to lines touching that junction, and so that each line node is connected to the two junction nodes corresponding to the junctions which that line touches.
- Produce likelihoods for the various node and junction labels in the final MRF.

4.2.1 Line and Junction Roles

Figure 6 illustrates the *roles* that lines may play relative to junctions. There are three possible roles, r_0, r_1, and r_2, which are assigned as shown in the figure. This assignment is easily done by measuring the angles between adjacent lines at a junction, as follows:

> r_0 is that line for which the clockwise arc distance to the next line is greatest, r_1 is that next line, and r_2 is the next line after that. If there are only two lines meeting at the junction, r_2 is null.

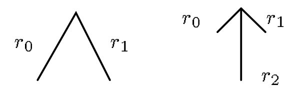


Figure 6: Line Roles for Different Junction Types

These roles serve to let us differentiate one line from another at a junction. This will be crucial when designing cliques for the MRF itself. Note that we do *not* assign junction labels to junctions at this point; that is done by the MRF. *The idea behind this is to give us enough information to solve the labeling problem without rigidly classifying a given junction prematurely.*

There are also roles relative to *lines.* The extraction mechanism assigns a role to each of the two junctions that a given line connects: these are j_0 and j_1. There is no particular significance to the numbering here; it is important simply to keep the two distinct. This, together with the other connectivity information described above, is enough to allow us to build the neighborhood graph for the MRF, which will be presented below.

4.2.2 External Evidence

There is one last thing the extraction mechanism must do: determine the evidence for each of the possible junction and node labels for each node in the MRF being built. The evidence is currently based on relatively *ad hoc* personal judgments of what would be appropriate likelihoods, given a particular feature of the image. The values given here have consistently yielded good results.

The label set for junctions is

$$L_{junction} = \{L, Y, Arrow, T\}, \tag{14}$$

In the determination of likelihoods for these labels, we examine two cases, depending on the number of lines at a given junction:

Two lines: In this case, the junction has to be an "L", so we set $P(O_s|L) = 0.9$, and set $P(O_s|\lambda) = 0.01, \lambda \in \{Y, T, Arrow\}$.

Three lines: In this case, we let α be the largest clockwise arc distance from any of the three lines to the next, and compute the likelihoods for each of the three-line labels as a function of α, as shown in Figure 7. We also set $P(O_s|L) = 0.01$.

The label set for lines is

$$L_{line} = \{+, -, \rightarrow\}, \tag{15}$$

i.e. lines can be labeled as either convex, concave, or occluding.

Note: It is important to point out that this line label set is not identical to the original Huffman/Clowes formulation. In particular, in the case of occluding edges, the original formulation marked which side of the edge the foreground of the object was on; this label set does not capture that distinction. Thus, we have a *collapsed label set*, and the solutions found are underspecified in the sense that they do not indicate what is being occluded by what. The decision was made to use this label set since it simplified the graph structure of the final MRF somewhat. Current work is directed at updating this system so as to capture the missing distinction.

Given this, it is straightforward to compute reasonable likelihood estimates for the line labels. For a given line, we let β be the fraction of the length of the line which borders the background;

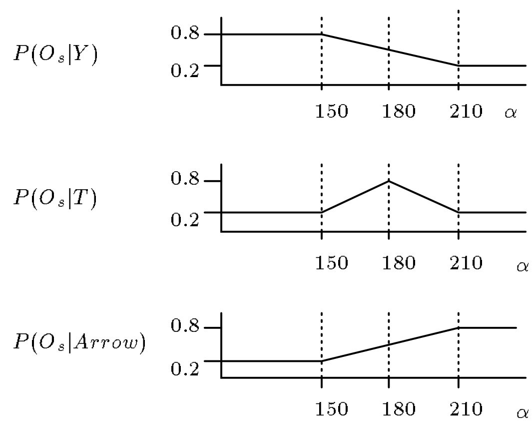


Figure 7: Computing Junction Likelihoods

---

this is computed during the line-tracing process described above. Then

$$P(O_s|+) = 1.0 - \beta,$$
$$P(O_s|-) = 1.0 - \beta, \text{and}$$
$$P(O_s|\rightarrow) = 0.5.$$

This reflects the fact that if the edge is either convex or concave, we know it will not border the background, while if it is an occluding edge, it may or may not.

4.3 Line and Junction Labeling

This brings us to the main focus of this report, namely the MRF responsible for line and junction labeling. As there are separate label sets for lines and junctions in the line and junction labeling MRF, this is in fact a *coupled* MRF.

We consider first the structure of the neighborhood graph for this MRF, and then the cliques used.

4.3.1 The Neighborhood Graph

Figure 8 should serve to give a feel for the interplay between junction nodes and line nodes in this coupled system. Figure 8(a) shows a portion of some trihedral object, with two junctions and four edges shown.

Figure 8(b) indicates which line nodes are neighbors of the two junction nodes, e and f. As described above, the structure of a junction node's neighborhood reflects the structure of the junction in the original image, in that the roles (r_0, r_1, r_2) are filled appropriately. For example, since the clockwise arc distance from B to A in (a) is greater than that from A to B, junction e's r_0 neighbor is B.

Figure 8(c) indicates which nodes are neighbors of line node A. Recall that junctions are assigned roles relative to lines, as well; junction e fills role j_0 relative to line A while junction f fills role j_1.

Note that line nodes also have, as immediate neighbors, the neighbors of neighboring junction nodes. Thus, the graph constructed by the extraction process also causes line node A to have, as

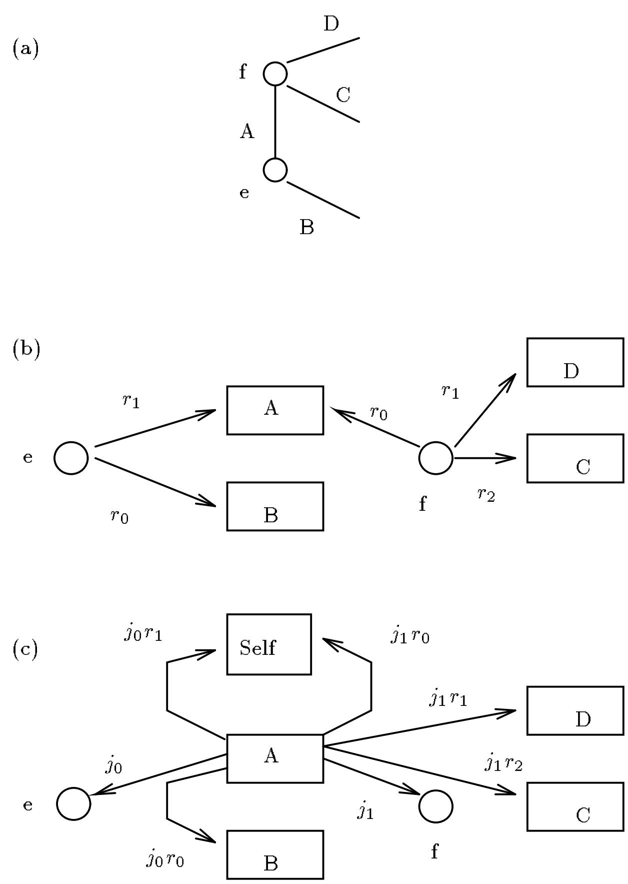


Figure 8: Neighborhood Relations

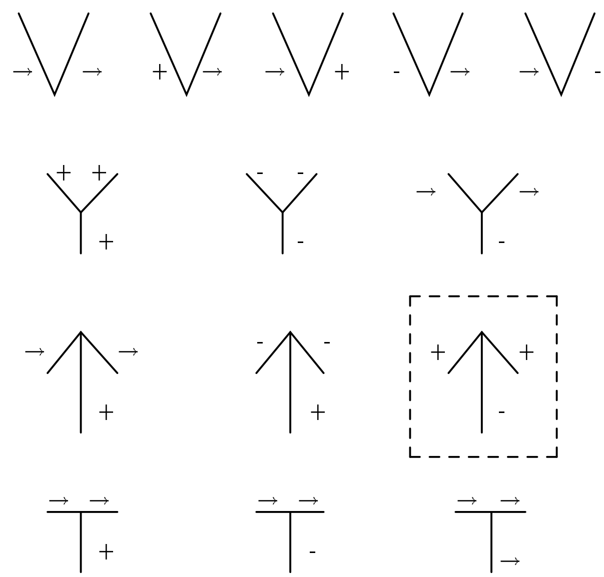


Figure 9: Junction Catalog (adapted from Huffman)

neighbors, each of the *line* nodes which play roles relative to A's junction neighbors. The neighbor filling the j_0r_0 role for A is that line which fills the r_0 role for A's j_0 neighbor. This is line node B.

Under this scheme, one would expect A's j_0r_1 neighbor to be A itself, since the r_1 neighbor of A's j_0 neighbor (e) is A. Instead, the fact that a particular *role* of A is filled by A itself is encoded by having the corresponding *neighbor* of A be a special node permanently labeled “Self”. I.e. the “Self” node denotes the fact that a node's neighbor is meant to be the node itself. This is done for reasons that will be made clear when discussing the cliques over this graph.

Since each line node has two junction neighbors, and since each of these junction neighbors has the original line node as a neighbor, the “Self” node is pointed to twice by each line node in the system.

4.3.2 Cliques Used in Line Labeling

Figure 9 presents the junction catalog which the cliques of this MRF embody. It has been adapted from the original, as we are currently not capturing the directionality of occluding edges.

Figure 10 presents three sample cliques from the set used here. The node being currently examined is marked by the symbol “?” in each of these. As is usual in the MRF formalism, we try to match each clique against subgraphs of the neighborhood graph.

Figure 10(a) expresses the knowledge that if the current line is in role r_2 relative to a junction which has been labeled “Arrow”, and if roles r_1 and r_0 of that junction are lines which have been labeled “+”, then it is favored[4] that this line node take on the label “-”.

Note how the “Self” node is used. We can check to see if the r_2 neighbor of the current node's j_0 neighbor is the current node itself, simply by checking to see if the label on the j_0r_2 neighbor is “Self”. It is crucial to be able to do this, since we need to be sure that the line node we are currently

[4] Recall equations 3 and 13. In general, a low value for a clique potential indicates that the corresponding clique is a relatively acceptable (probable) configuration, while higher values indicate less likely configurations.

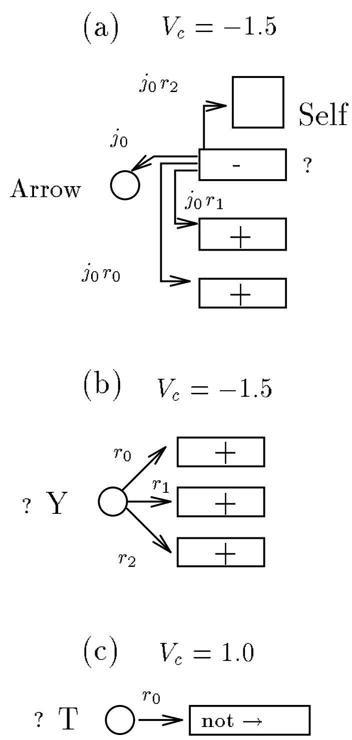


Figure 10: Cliques for Line Labeling

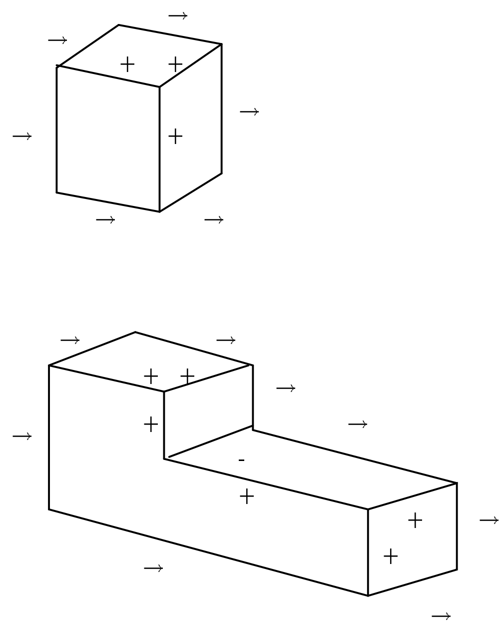


Figure 11: Example Results

examining is in fact the middle line in an arrow junction. It is, if the j_0 neighbor is labeled "Arrow", and the j_0r_2 neighbor is labeled "Self".

Referring back to Figure 9, and focusing on the junction highlighted by a dashed outline, we find that the clique corresponds to the assertion that the middle branch of the arrow should be labeled "-" if the junction has been labeled as an arrow, and if the other two branches of the junction have been labeled "+". Since this clique covers only junctions which are in relation j_0 to the line in question, we require a similar clique to take care of the corresponding case when the junction fills role j_1.

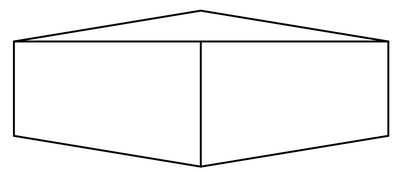
While considerations of space preclude the inclusion here of a complete listing of the cliques used by the system, most are of the type shown in (a), and encode very specific pieces of knowledge from the catalog, regarding what line labels are acceptable in what configurations.

Figure 10(b) encodes the knowledge that if all three lines touching a junction are "+", it is appropriate to favor labeling the junction "Y". Conversely, (c) encodes the knowledge that if the line filling the role r_0 for some junction is labeled something other than "$\rightarrow$", the junction should probably not be labeled "T".

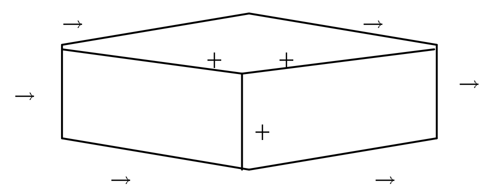
It is the inclusion in the system of cliques of this latter sort, governing the appropriateness of particular junction labels, that makes this work something more than just an MRF implementation of a standard line-labeling algorithm. Through these cliques, the system is able to perform junction labeling which is not just a direct reflection of the evidence, so that ambiguous or misleading evidence can be overcome, resulting in a globally consistent solution.

5 Results

Figure 11 presents correct labelings produced by the system for a pair of trihedral objects. The



(a)



(b)

Figure 12: Overriding Misleading Evidence

first of the two objects is the cube shown in Figure 1. Input to the coupled MRF was constructed by hand for the second of the two objects shown.

In both cases, the number of node updates required was equal to one per node. Recall that the HCF algorithm first updates those nodes which have strong evidence in favor of a particular label. Thus, "L" junctions and occluding edges tend to be marked first, as there is always strong evidence for each of them.[5] These decisions then influence the other choices, eventually resulting in a complete labeling.

More interestingly, Figure 12 presents a situation in which misleading evidence (the apparent "T" junction in (a)) is outweighed by constraints placed on the junction label by the adjacent line labels. Thus, the solution arrived at in (b) is one in which what looked like a "T" junction has been labeled as a "Y" junction.[6] This extreme case indicates the system's ability to overcome the problem of perceptual similarity of different junction types, since in this example, a junction which had evidence identical to that for a "T" was labeled as a "Y" since that was the node label which resulted in an overall labeling with maximum *a posteriori* probability.

As one might expect, the "Y" junction is the last to be labeled, as it is the identity of the three convex lines that touch it that force it to take that label (recall Figure 10(b) and (c)). This is in contrast to the first figure in Figure 11, in which the "Y" had strong evidence, and was labeled relatively early.

[5] The presence of only two lines at a junction is strong evidence for an L, and the presence of a stretch of background bordering an edge is strong evidence for an occluding edge (though an occluding edge need not border the background).

[6] Input to the final MRF was constructed by hand for this case as well.

6 Conclusions

The problem of perceptual similarity of different junction types, in producing a line labeling from a real image, is addressed through a system which performs both line labeling and junction labeling, in a coupled probabilistic system. Lines and junctions are first detected in the image, and then extracted and used to build a graph for a coupled MRF. This MRF then produces the labeling.

Extensions to this work are currently under consideration. Among the extensions being looked into are (a) marking occluding edges for directionality, which will require some minor modifications of the neighborhood graph structure for the MRF, and (b) using other junction catalogs, covering more than simply trihedral objects. In particular, the junction catalog of [Malik, 1985] has features that seem to require some of the abilities of this system, as we have seen. This would thus be an appropriate direction for extension.

References

[Chou and Raman, 1987] Paul Chou and Rajeev Raman, "On Relaxation Algorithms Based on Markov Random Fields," Technical Report 212, Computer Science Department, University of Rochester, 1987.

[Clowes, 1971] M.B. Clowes, "On Seeing Things," *Artificial Intelligence*, 2:79–116, 1971.

[Cooper, 1989] Paul Cooper, "Parallel Object Recognition from Structure," Technical Report 301, PhD. thesis, Department of Computer Science, University of Rochester, July 1989.

[Cross and Jain, 1983] G. R. Cross and A. K. Jain, "Markov Random Field Texture Models," *IEEE PAMI*, 5(1):25–39, January 1983.

[Geman and Geman, 1984] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE PAMI*, 6(6):721–741, November 1984.

[Huffman, 1971] D. A. Huffman, "Impossible Objects as Nonsense Sentences," *Machine Intelligence*, 6:295–323, 1971.

[Malik, 1985] Jitendra Malik, "Interpreting Line Drawings of Curved Objects," Technical Report 1099, PhD. thesis, Department of Computer Science, Stanford University, December 1985.

[Waltz, 1975] D. Waltz, "Understanding Line Drawings of Scenes with Shadows," In *The Psychology of Computer Vision*. McGraw-Hill, 1975, edited by P. H. Winston.