Proceedings of the Seventeenth International Conference on Machine Learning, 567–573.
San Francisco, CA: Morgan Kaufmann

An Initial Study of an Adaptive Hierarchical Vision System

Marcus A. Maloof MALOOF@CS.GEORGETOWN.EDU
Department of Computer Science, Georgetown University, Washington, DC 20057-1232, USA

Abstract

We describe an empirical study of an adaptive, hierarchical vision system. Using a simple vision task requiring both low-level and high-level processing, we examined how three schemes of feedback for on-line learning affected the true positive rate, the number of instances used for learning, and the need for user feedback. The first scheme used for learning those instances for which the user provided feedback. The second used all instances, assuming that no feedback meant correct classification. In the end, a hybrid scheme with each method at different levels yielded the best results, showing that more examples for learning significantly improved the true positive rate of the classifiers at the lower level, but not at the higher level. Furthermore, this hybrid method did not increase the need for user feedback.

1. Introduction

Hierarchical vision systems (Mohan & Nevatia, 1989) work by repeatedly grouping visual constructs into higher-level constructs and selecting the most promising to pass up to other modules in the hierarchy. This process starts at the image level, where the system surveys local information, such as pixel intensities, and proceeds upward, taking into account more global types of information, such as spatial relationships, until constructing a candidate description of the object in the image. This approach is useful for objects that are difficult recognize reliably by simply analyzing the shape of their occluding contour.

Of particular interest are the mechanisms in the hierarchy that select the most promising constructs for further processing. Traditionally, researchers have used a variety of handcrafted, deductive methods for construct selection, which often prove brittle because of our inability to survey the evidence necessary for robust behavior. A greater problem is that these mechanisms are static, meaning that the system, once deployed, cannot adapt or change. People often supervise or interact directly with such systems, so this suggests a role for on-line learning with the aid of user feedback.

The goal of our research is to develop an adaptive architecture for hierarchical vision systems that, with the aid of user feedback, uses learning to improve both the mechanisms responsible for selecting the most promising constructs and the underlying vision algorithms responsible for producing those constructs. Our aim is to build an experimental system based on the architecture for a complex vision task, such a building detection.

In this paper, we describe an adaptive architecture for hierarchical vision systems that uses on-line learning methods for construct selection. We extend our previous work (Maloof, Langley, Sage, & Binford, 1997; Maloof, Langley, Binford, & Nevatia, 1998) by using on-line learning at multiple levels of a hierarchical vision system. For a simple but representative vision task, we empirically studied how three schemes of feedback affected three aspects of performance: the true positive rate, the number of examples used for on-line learning, and the number of times the user provided feedback.

The first scheme, called *partial instance feedback*, entailed learning from instances for which the user provided feedback, while the second, called *full instance feedback*, involved learning from all instances under the assumption that no feedback meant correct classification. These are similar to methods used in IB2 and IB1 (Aha, Kibler, & Albert, 1991), respectively, although we did not use instance-based learning.

We anticipated that full instance feedback would yield higher true positive rates and would thereby reduce the need for user interaction, as the system would make fewer mistakes. However, results demonstrated that, at the higher level of the hierarchy, the full feedback method actually reduced the true positive rate, a finding that we attributed to a skewed or imbalanced data set. In the end, a hybrid scheme using the full feedback method at the lower level and the partial feedback method at the higher level yielded the best results.

The organization of the paper is as follows. In the next section, we discuss the vision task of detecting blasting caps in X-ray images. We then describe our adaptive architecture for hierarchical machine vision systems. This is followed by a section detailing our experimental study: the data, the learning method, the experimental design, and an analysis of the results. We conclude with a brief discussion of related work and with thoughts for future directions.

2. The Vision Task

For this study, we selected the vision task of detecting blasting caps in X-ray images. We chose this task because it is relatively simple and is representative of a much larger class of tasks that requires both low-level processing (e.g., edge detection) and high-level processing (e.g., spatial and geometric reasoning). Building detection in overhead imagery is an example of a more complex task in which we need low-level processes to extract edges, intermediate-level processes to group lines into corners, and high-level processes to reason about the geometric properties of parallelograms corresponding to rooftops (e.g., Lin & Nevatia, 1996).

As shown in Figure 1, blasting caps, when X-rayed, produce two discernible regions: the region corresponding to a heavy-metal explosive at the center of the cap, and the region produced by the metal tube of the cap. Detecting both of these regions is necessary, but not sufficient, as the spatial relationship between these two regions is also important. Therefore, the vision system must also establish that the blobular region is contained within the rectangular region and is roughly near its center. This suggests a simple recognition hierarchy, which we will discuss in the next section.

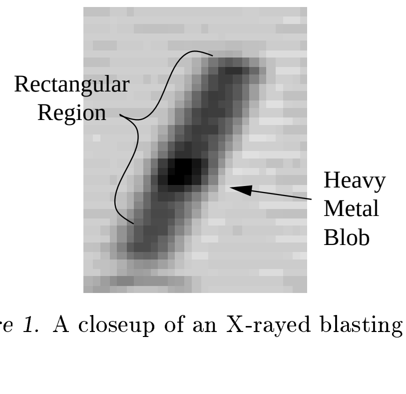

Figure 1. A closeup of an X-rayed blasting cap.

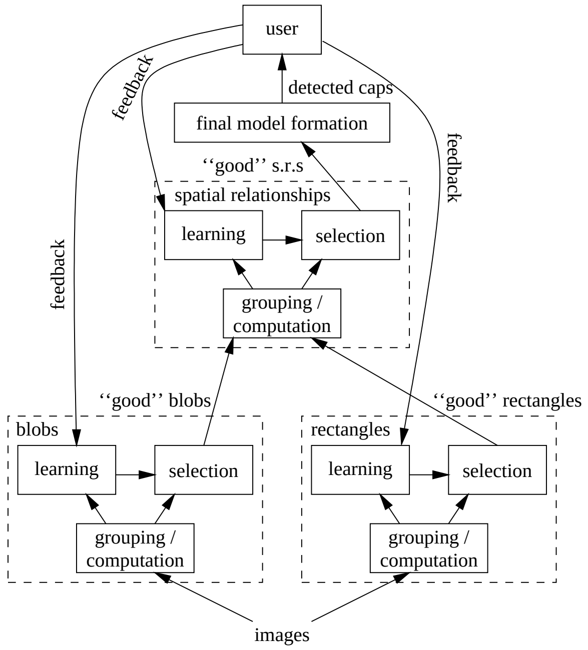

Figure 2. An adaptive, hierarchical vision system for detecting blasting caps.

3. An Adaptive Vision Architecture

A hierarchical approach to this problem involves detecting blobular regions, detecting rectangular regions, and, from these, detecting that the proper spatial relationship exists, as pictured in Figure 2. Each module in the hierarchy works similarly, by taking lower-level constructs, grouping them, and computing a set of predefined features. A selection process uses concept descriptions to select the most promising visual constructs for further processing based on these predefined features. These selected constructs, and possibly a subset of the computed features, serve as input to modules higher in the hierarchy.

This process continues until the system constructs a representation of the object in the image. The user reviews the detected object and may provide positive or negative feedback, which propagates to the modules in the hierarchy and starts on-line learning processes that update the concept descriptions for construct selection. This continues throughout the life of the vision system.

Ideally, we would like for the user to provide feedback only about the final object description. In practice, this is difficult because a lower-level visual construct can be part of both a valid and an invalid object description. For example, with a building detection task, a linear feature corresponding to an edge of a building is a part of a valid description of the building, but it is also a part of an invalid description of the area adjacent to the building. We revisit this issue when we discuss future work but, in the next section, turn to the details of our experimental study.

4. Experimental Study

In this initial study of learning at multiple levels in a hierarchical vision system, we investigated three schemes of feedback. We assumed that the user would provide feedback for every misclassified construct. In practice, such a scheme would not be realistic due to the potential burden of labeling on the user, but we were interested in evaluating how these schemes of feedback affected the true positive rate, the number of examples used for learning, and the need for user feedback.

4.1 Images and Training Data

We derived the data for this study from 28 X-ray images of airport luggage containing blasting caps. With the aid of an expert, we identified regions of interest in the images and computed a variety of intensity and shape features (Jain, 1989). From the available regions, we randomly selected 102 positive and 102 negative examples of blobular and rectangular regions. We suspected that not all of the features were relevant for detection, so we selected a subset using a t-test at $p < .01$.

Eleven relevant attributes remained for the blobular regions: area, perimeter, compactness, maximum and standard deviation of intensity in the region, the eccentricity and lengths of the major and minor axes of a fitted ellipse, the length and width of an oriented bounding box, and the ratio of this length and width. Eight relevant attributes remained for rectangular regions: compactness, the eccentricity and length of the major axis of a fitted ellipse, the width of an oriented bounding box, and the average, minimum, mode, and standard deviation of intensity in the region. To capture the spatial relationship between the blobular region and the rectangular region, we used three features computed from the regions' centroids: Euclidean distance, and the horizontal and vertical components of this distance. With the exception of the minimum, maximum, and mode of the intensity, all of these measures were real-valued.

4.2 Experimental Method

For the learning process of each module in the hierarchy, we used an on-line version of naive Bayes. Since all of the attribute values were numeric, we stored the sum and the sum of the squares of each attribute's values, assumed a normal distribution, and estimated the conditional probability of a numeric value from the mean and variance of an attribute's values.

Assuming that the user provided feedback for every misclassified instance, as our independent variable, we evaluated three methods of feedback. Partial instance feedback (henceforth "partial") used only the instances for which the user provided feedback for on-line learning, whereas full instance feedback (henceforth "full") used every instance for on-line learning by assuming that instances without feedback were classified correctly. Prompted by results from an earlier experiment, we also evaluated a hybrid scheme that used the full method at the lower level and the partial method at the higher level.

For dependent variables, we measured the number of times the user provided feedback for a misclassified instance, the number of instances used for on-line learning, and the true positive rates for detecting blobular regions, rectangular regions, and the spatial relationship that exists between these regions. We anticipated that, since the full method would use more instances for learning, it would reduce the number of user interactions, as compared to the partial method.

The experiment for this study consisted of forty runs. Each began by randomly selecting thirty examples from the blobular class to serve as a test set. We also selected thirty examples from the rectangular class. However, because we must test detection on whole objects, not just on parts of objects, for each positive example of a blobular region, we retrieved its corresponding rectangular region. There was no such requirement for negative examples, for they do not constitute valid objects. From the thirty examples in the test sets for the blobular and rectangular regions, we derived the test set for the spatial-relationships class by computing the distance measures from each region's centroid.

We then constructed an initial training set of six examples—two from each class, one positive, one negative—by the same method for constructing the test sets, and used naive Bayes to induce three initial concept descriptions, one for detecting blobular regions, one for detecting rectangular regions, and one for detecting the spatial relationship that exists between them.

At this point, the on-line portion of the experiment began and involved randomly selecting an instance from the blobular regions and classifying it, and randomly selecting an instance from the rectangular regions and classifying it. If the system classified both regions

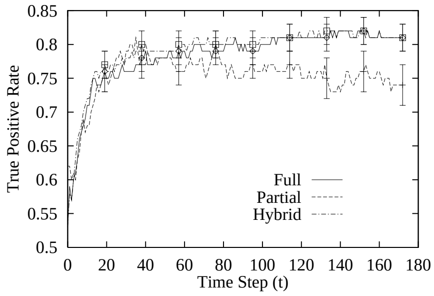

Figure 3. True positive rate for detecting blobs.

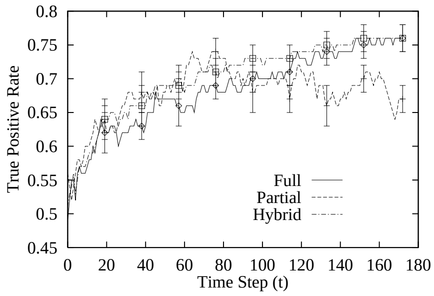

Figure 4. True positive rate for detecting rectangles.

as positive, then we constructed an instance of the spatial-relationships class by computing the distance measures from each region's centroid. The system then classified this instance as either positive or negative.

If any of the instances were misclassified, then we assumed that the user provided feedback. In the case of the partial feedback scheme, we used only the instances for which the user provided feedback for learning (i.e., those that were misclassified, but with their correct labels). For the full feedback method, we used all three instances for on-line learning, those that were correctly classified and those that were misclassified, again, with their correct labels. As stated previously, for the hybrid scheme, we used the full method at the lower level and the partial method at the higher level.

After using the instances for on-line learning, we evaluated each new concept description on the examples in their corresponding test set, computing the true positive rates for blob detection, for rectangle detection, and for spatial-relationship detection.

We repeated this on-line procedure until processing all of the examples in the training sets. At the end of experiment, we averaged the performance metrics over the forty runs and computed 95% confidence intervals, which we plotted as performance curves. We discuss these results in the next section.

4.3 Experimental Results

In our experiment, we examined three methods of feedback—the full feedback method, the partial feedback method, and the hybrid method—and measured the number of instances used for on-line learning, the number of times the user provided feedback, and the true positive rates for detecting blobular regions, rectangular regions, and the spatial relationship that exists between these regions.

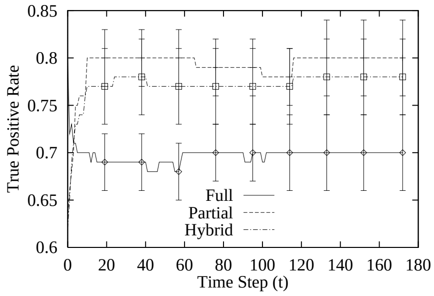

Figure 5. True positive rate for detecting spatial relationships.

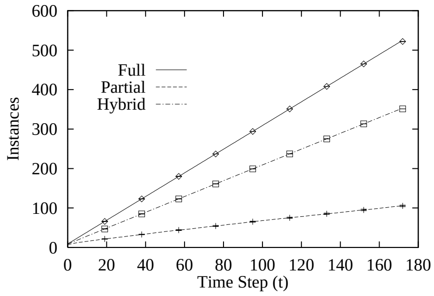

Figure 6. Number of instances used for on-line learning.

Because the full method used every instance for learning, as we anticipated, the true positive rate for this method was higher than that of the partial method, as Figures 3 and 4 show. Furthermore, since the hybrid scheme used the full method at the lower level, its learning curves for both tasks were virtually identical to those of the full method.

However, this intuition did not hold for the true positive rate for detecting the spatial relationships, as Figure 5 indicates. The reverse of what we expected occurred, as the true positive rate on this task was higher for the partial method than for the full method. A similar result from an earlier experiment prompted us to investigate the hybrid method, which achieved a true positive rate comparable to the partial method.

Turning to the number of instances used for on-line learning, shown in Figure 6, we see that the full feedback scheme, since it used all instances for learning, produced a line from the origin to 522, which was the total number of instances used for training over the three classes. The partial scheme used notably fewer instances, processing only 105 ± 2 instances over the three classes. Finally, the hybrid scheme processed 351.18 ± 0.12 instances.

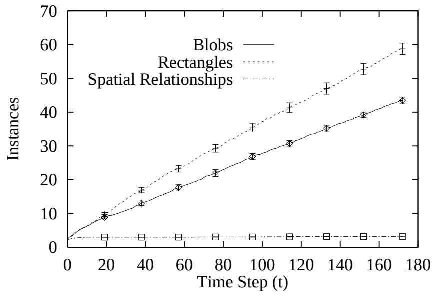

Figure 7. Distribution by class of instances used for the partial feedback method.

However, this graph does not show the distribution of the instances among the classes. In the case of the partial method, as shown in Figure 7, the majority of the instances were from the rectangle class. This was expected since the true positive rate for the rectangle class was lower than that for that blob class, meaning that the former classifier made more mistakes, which, in turn, resulted in more instances being used for online learning. Surprising was the small number of instances from the spatial-relationships class used for online learning, which was 3.15 ± 0.1 at $t = 172$. The hybrid method also used an equally small number of instances for learning: 3.17 ± 0.12, also at $t = 172$.

Finally, in Figure 8, we see the number of times the user provided feedback. We expected that the full feedback scheme would reduce the need for user feedback. Intuitively, the full feedback scheme would entail learning from more examples, which would yield a higher true positive rate, resulting in fewer mistakes and, consequently, fewer times the user would have to provide feedback. But, as the graph indicates, the difference in the number of interactions for the full method versus the partial was only seven examples (99.35 ± 2.02 versus 92.45 ± 2.02, respectively). The hybrid method required 91.65 ± 1.79 interactions, which was not significantly different from that of the full method.

In terms of overall accuracy, because the system made three separate decisions at a true positive rate of roughly 0.8, the true positive rate for object detection was quite low. The partial method ultimately achieved a true positive rate of 0.38 ± 0.02, the full method achieved 0.40 ± 0.03, and the hybrid method, 0.46 ± 0.03. However, to put these results in context, a default classifier configured similarly for this task achieves a true positive rate of 0.125.

4.4 Analysis

In examining the results of the previous section, we were most intrigued by the relationship between the true positive rate of the classifier for spatial relationships and the number of instances used for learning. Intuitively, the full method should have yielded a classifier with a higher true positive rate. Yet, as the results demonstrated, learning from more examples did not translate into a higher true positive rate.

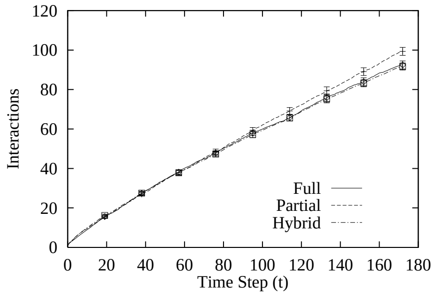

Figure 8. Number of user interactions.

Several phenomena could account for this result, such as a poor representation of spatial relationships, irrelevant attributes, inductive bias, or overfitting. We ruled out each of these possibilities through an analysis of the descriptive statistics of the training data and of the concept descriptions produced by naive Bayes.

In the end, we concluded that a skewed training set caused the problem. Originally, we attempted to control for problems associated with skew by randomly selecting balanced training sets. Recall that we began by selecting 204 blobular and 204 rectangular regions with an equal distribution of positive and negative examples. However, between blobular and rectangular regions, there are 102 valid spatial relationships and roughly $2^{102} - 102$ invalid relationships.

Indeed, upon further inspection of the results, we found that, on average, the spatial-relationships classifier at $t = 172$ for the full feedback method had been derived from 1.4 ± 0.2 positive examples and 172.6 ± 0.2 negative examples. In contrast, the spatial-relationships classifier for the partial feedback method, also at $t = 172$, had been derived from 1.0 ± 0.0 positive example and only 2.25 ± 0.4 negative examples. We concluded that the partial method was less susceptible to problems associated with skew because it learned only from misclassified examples.

Accuracy on this task did not suffer from the lack of examples because the decision regions were easily separable, making detection relatively simple. As an example, for the positive instances of the spatial-relationships class, the mean of the distance attribute was 7.3 with a standard deviation of 3.7, whereas, for the negative instances, these measures were 196.2 and 96.8, respectively. We will return to the issue of skew when we conclude, but, in the next section, we discuss related work.

5. Related Work

There are many bodies of related research, but we will cite a few examples of work involving hierarchies and learning. Connell and Brady (1987) describe a model-based vision system that generates semantic networks representing commercial aircraft detected in overhead images. Such networks for individual planes served as training examples for a learning process that produced a generalized description of the aircraft. In this work there was a strong separation between the vision process and the learning process. Consequently, learning was restricted to generalizing only fully formed object descriptions. There was no way to improve the intermediate representations or the vision mechanisms.

An approach complimentary to ours is Draper's, who investigated reinforcement learning to derive the hierarchies for object recognition using the tasks of rooftop detection (Draper, 1996) and pose estimation of eye-level buildings (Draper, 1997). This work was described as an off-line approach, so we view it as method for learning the recognition hierarchy that our system would refine in an on-line setting with the aid of user feedback.

Although acquired through reinforcement learning, Draper's hierarchies are different from those used in hierarchical reinforcement learning (e.g., Dietterich, 1998), which decompose a task into a sequence of subtasks; however, at a given level in these hierarchies, there is a temporal ordering of subtasks. At a given level in ours and in Draper's, there is no such ordering, and we could execute tasks in parallel.

Our hierarchies share some similarities with hierarchical mixtures of experts (Jordan & Jacobs, 1994), which is a tree-structured approach for supervised learning. Information flows bottom-up, as in ours, but each module uses all of the training examples for learning. In ours, each module learns a different classification task and could consist of different learning and performance elements.

Finally, our hierarchies are also similar to causal networks (e.g., Pearl, 1988). Nodes in these networks represent random variables, while links between nodes denote causal relationships. Several researchers have investigated techniques for learning both the structure and parameters of such networks (e.g., Heckerman, 1995). We plan to examine how techniques for learning causal networks with hidden variables might apply to our learning task, but we leave this for future work, which we discuss further in the final section.

6. Concluding Remarks

In this paper, we have described an adaptive architecture for hierarchical vision systems. Using a simple vision task, we examined three methods of feedback and measured the number of user interactions, the number of instances used for on-line learning, and the true positive rates of each classifier in the hierarchy. The true positive rate for spatial-relationships detection did not match our expectations because of problems associated with data skew.

The fact that we encountered the problem of data skew was not surprising. Indeed, we have encountered it in our earlier studies involving a rooftop detection task (Maloof et al., 1997). However, we did not expect that the partial feedback method would filter such a large portion of the negative examples, thus lessening the effects of skew. Naturally, we must be cautious about this conclusion because, as we discussed previously, this was a relatively simple task, and, on a more complex task, performance may suffer from the lack of examples. Nevertheless, in this context, these feedback mechanisms may be alternatives to, say, sampling or cost-sensitive learning for coping with skew.

For the future, we plan to investigate methods that will reduce the need for user feedback. In particular, we plan to examine methods that will use negative feedback on the final object description to infer the most likely set of labels for the constructs distributed throughout the hierarchy. The results from this study should provide a basis for comparison.

In this study, we assumed that the user provided feedback for every misclassified construct, which may be impractical for more complex domains because of the large number of constructs involved. We may be able to reduce labeling by first attempting to relabel every construct of a false-negative or a false-positive object description. As we stated previously, this will not always suffice since a construct can be part of both a valid and an invalid object description. In the case of such conflicts, the system could present only those constructs in conflict to the user for feedback. This suggests several empirical studies to determine trade-offs between accuracy and user interaction.

Finally, with on-line learning algorithms in each module of the hierarchy, in addition to refining the knowledge for selecting visual constructs, we plan to explore how to adjust the parameters of the vision algorithms responsible for producing these constructs. We anticipate that these directions will move us closer to our ultimate goal of developing adaptive vision systems.

Acknowledgements

This research was conducted in the Department of Computer Science at Georgetown University in Washington, DC. The author thanks the anonymous reviewers for constructive and helpful comments.

References

Aha, D., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, *6*, 37–66.

Connell, J., & Brady, M. (1987). Generating and generalizing models of visual objects. *Artificial Intelligence*, *31*, 159–183.

Dietterich, T. (1998). The MAXQ method for hierarchical reinforcement learning. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 118–126). San Francisco, CA: Morgan Kaufmann.

Draper, B. (1996). Learning grouping strategies for 2D and 3D object recognition. In *Proceedings of the Image Understanding Workshop* (pp. 1447–1454). San Francisco, CA: Morgan Kaufmann.

Draper, B. (1997). Learning control strategies for object recognition. In K. Ikeuchi & M. Veloso (Eds.), *Symbolic visual learning* (pp. 49–76). New York, NY: Oxford University Press.

Heckerman, D. (1995). A Bayesian approach to learning causal networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 285–295). San Francisco, CA: Morgan Kaufmann.

Jain, A. (1989). *Fundamentals of digital image processing.* Upper Saddle River, NJ: Prentice-Hall.

Jordan, M., & Jacobs, R. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181–214.

Lin, C., & Nevatia, R. (1996). Building detection and description from monocular aerial images. In *Proceedings of the Image Understanding Workshop* (pp. 461–468). San Francisco, CA: Morgan Kaufmann.

Maloof, M., Langley, P., Binford, T., & Nevatia, R. (1998). Generalizing over aspect and location for rooftop detection. In *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision* (pp. 194–199). Los Alamitos, CA: IEEE Press.

Maloof, M., Langley, P., Sage, S., & Binford, T. (1997). Learning to detect rooftops in aerial images. In *Proceedings of the Image Understanding Workshop* (pp. 835–845). San Francisco, CA: Morgan Kaufmann.

Mohan, R., & Nevatia, R. (1989). Using perceptual organization to extract 3-D structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*, 1121–1139.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* San Francisco, CA: Morgan Kaufmann.