A New Approach to Early Sketch Processing

Sonya Cates and **Randall Davis**
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street
Cambridge, MA 02139
{sjcates, davis}@csail.mit.edu

Abstract

Early processing of sketches is a requirement common to many pen based systems. The task is difficult because of variations between sketches from different users and in different domains, and because of ambiguities that arise within even slightly messy sketches, especially when the input is allowed to be unconstrained.

We propose a graphical model based approach to early sketch processing. Small areas of a sketch corresponding to features such as corners and straight segments are considered individually, and a likely labeling for such features is found by incorporating some context in order to improve on labels computed with only local information. Results from applying this approach to the problem of detecting corners show an improvement.

Introduction

Most pen based systems perform some type of early processing of sketched input. Often the extracted information, such as the description of a shape, the time it was drawn, and the position on the page, serves as the input to higher level processing, such as recognition, inference, or beatification.

Several approaches to early processing are used in current sketch systems. Hse, Shilman, and Newton (2004) determine shape by fragmenting a stroke into straight and curved segments and then matching the segments to a template. ScanScribe (Saund *et al.* 2002) also segments the input into curved and straight segments as part of a pre-processing stage. At the first level of processing, Gross and Do (2000) extract features including pen path and aspect ratio, which are then matched to templates.

Though often intuitively simple, the early processing or pre-processing stages can pose difficult problems. There is often a trade-off between making a flexible system that allows the user to draw in the most unconstrained way possible and making an early processing system that performs consistently and accurately in a noisy sketching environment. For example, Palm Pilot Graffiti is robust enough for millions of users with very little training, but the trade-off is very limited input. The ideal would be a system that is both flexible and accurate.

Human interpretation of shapes, such as those in Figure 1, is based on a variety of features, including curvature, sharp direction change and whether or not the shape as a whole is closed. These features occur at a variety of scales. Some, like the presence of a corner, describe a very small area. Some, like whether the shape is closed, describe the entire stroke, and some, like whether a portion of a stroke is straight, arise at an intermediate scale. In order to create a very flexible system, we use features at different scales, such as those described, as our vocabulary to describe a shape.

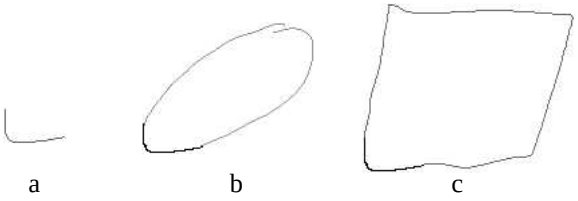


Figure 1: An example of local ambiguity. Considered out of context, the segment shown may equally well be the side of the ellipse and the corner of the square.

Quadrilaterals can be detected by testing whether the shape is closed, it has four corners, and the segments between the corners are straight. Rather than having a set of predefined shapes, a stroke is initially described by its features in this way. We take this approach in order to add new shapes easily and to give the user the flexibility to use undefined shapes in a sketch.

Classifying these single features, some of which are very localized, can be problematic. Sketching styles often vary considerably, making it difficult to construct a general system capable of correctly classifying all instances of corners, parallel lines, straight lines, etc., across all users and all domains. Yet humans can perform this task consistently without extensive domain knowledge and without prior knowledge of an individual's drawing style. Furthermore, because sketches are almost never drawn perfectly, they often contain internal inconsistencies, i.e., the same element may have a different meaning in different parts of a sketch. As a result, no simple function can correctly classify all features, even for a single sketch. Yet humans usually agree on the intention of a local feature in spite of ambiguities. For example, in Figure 1, nearly identical partial strokes require different interpretations. We form a different interpretation for the shapes (b and c), even though it is impossible to distinguish between the side of the ellipse and the corner of the

Appeared in *AAAI 2004 Symposium on Making Pen-Based Interaction Intelligent and Natural, 2004.* pp.29-34.

square with only local information (a). This example illustrates the type of local ambiguity that is common in even relatively neat drawings. A template-based approach could easily distinguish between the ellipse and the square, but as we also aim to preserve the adaptability of a more general approach, a local unconstrained interpretation is preferred. A goal of this work is therefore to incorporate some context while continuing to identify features individually.

This paper presents an application of Markov random fields (undirected graphical models) and belief propagation (an inference algorithm based on message passing), in which rough, initial interpretations of local features are improved with evidence from nearby areas. We report here only the application of this approach to the problem of locating corners. This problem was chosen because it is a common source of error. We believe our approach may also be useful in resolving other ambiguous elements in sketches. MRFs with belief propagation have been applied to problems in vision, such as for locating text and shapes in images (Zhang & Chang 2004), (Coughlan & Ferreira 2002). Also, Roth and Yih (2002) use a similar inference method for identifying relationships and entities in text understanding.

In our model, each stroke in a sketch is represented as an MRF, with each node in the MRF representing an area identified as a possible corner or an area in between two possible corners. Connections between nodes establish the context for an area. Inference involves functions that represent the compatibility of the evidence associated with a single node or set of nodes and a particular labeling. We define these compatibilities based on qualitative principles.

Constructing the Graphical Representation of a Sketch

To construct its graphical representation, each stroke in a sketch is represented separately as an MRF. Figure 2 shows a stroke and its associated graph. Possible corners are found with a very weak classifier that finds any area along the stroke with direction change above a threshold. In the stroke in Figure 2, the possible corners are labeled by c_1, c_2, and c_2. Given a low enough threshold, this method will find all corners, but generally also returns as many false positives as true corners. Each possible corner is represented as a node with an associated random variable, which may have one of two labels $corner$ and $not\ corner$. The segments of the stroke between the possible corners are also represented as nodes.

More generally, the sketch only needs to be divided into relevant sections; rather than considering each stroke individually, other measures, such as distance or time could be used. We have defined two types of nodes, corresponding to possible corners and the segments in between, but the particular features that are relevant depend on the problem.

Nodes are connected according to the following rules:

1. all nodes of the same type are connected, forming a complete subgraph of order n, if there are n instances of a particular type of node
2. a node is connected to the closest nodes of different types

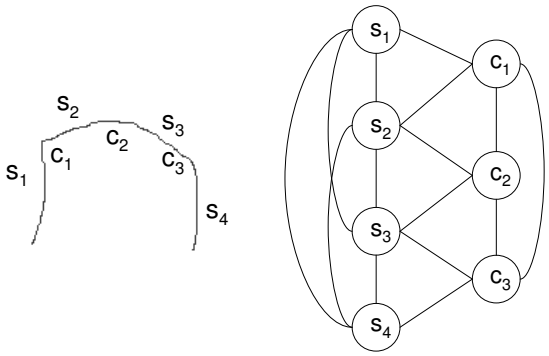


Figure 2: A stroke and its corresponding graph. Possible corners are labeled c_1, c_2, and c_3. Areas between the possible corners are labeled s_1,s_2, s_3, and s_4. Possible corners are found by selecting areas of the stroke with a direction change above a threshold.

We define the closest nodes as those that represent adjacent areas in the same stroke, but again, another measure involving distance or time could be used as well. As a result, for the problem of detecting corners, all nodes representing possible corners are connected to each other, all nodes representing segments between possible corners are connected to each other, and a node representing a possible corner is connected to the nodes for the segments on either side of it. For example, in Figure 2, corner c_1 is connected to the other possible corners, c_2 and c_3, and to adjacent segments, s_1 and s_2. The neighbors of a node in the graph directly influence its interpretation and may be thought of as the context for the node.

Markov random fields are most commonly used as pairwise MRFs, meaning that the largest clique is of size two. In our representation, however, the graph will have cliques with size equal to the number of each type of feature. Inference on such a graph is more complicated and time consuming. We found, however, that due to the restricted size and structure of the graphs in our formulation, approximate inference may still be performed in real time. The benefit we find from using more connected (rather than pairwise) graphs is that we can define the compatibility functions more intuitively, defining functions that describe the relationship of a group of nodes, rather than only pairwise interactions. Zhang and Chang (2004) make a similar argument for the use of higher order models.

Compatibility Functions

We define functions for each clique in the graph and for each individual node. The functions are a measure of compatibility between a particular labeling and the underlying data, and may be thought of as a measure of certainty for a particular labeling of a single node or a set of nodes. For a single node that has two possible labelings, for example a possible corner that may be labeled as $corner$ or $not\ corner$, we compute a compatibility for each label with observations made of the data. For a group of nodes forming a clique,

we compute a compatibility for all possible labelings of the group. These multi-node functions convey context.

In our model for finding corners, we define four functions: one for single nodes, one for all the nodes of the same type, one for a corner and its surrounding segments, and one for a segment and its surrounding corners.[1]

Single node functions

For the problem of detecting corners we define two types of nodes: those representing possible corners and those representing areas between possible corners. In both cases the single node function is computed by combining measurements on the raw data, including arc length, distance between the endpoints, direction change, and speed. We determine the weights for each measurement by performing logistic regression on a small labeled data set. The single node function is defined by:

$$\psi_{c_i}(x_{c_i}, y_{c_i}) = (1 + exp(w_0 + \sum_j w_j f_j(y_{c_i}))^{-1}$$

where ψ is the compatibility between a label x_{c_i} and an observation y_{c_i}. The w_j's are weights, and $f_j(y_{c_i})$ is the j^{th} measurement. For segments, $\psi_{s_i}(x_{s_i}, y_{s_i})$ is defined similarly and represents compatibility with the labels $straight$ and $not\ straight$.

These functions can be used alone to determine a labeling for the features, independent of context; we use them later as a baseline for comparison.

Functions of the same node type

We next define a compatibility for a labeling of all the possible corners given observations and a compatibility for a labeling of all the connecting segments. Intuitively, similar things should be given the same label, and different things given different labels. The group compatibility function conveys this heuristic. The group is split according to the labeling under consideration. For example, to find the compatibility of the possible corners in Figure 2 with the labeling $\{c_1 = corner, c_2 = not\ corner, c_3 = corner\}$ subgroups $\{c_2\}$ and $\{c_1, c_3\}$ are formed. Then the distance between the subgroups and the average distance from the mean in each subgroup are compared. It is defined by:

$$\psi_{c_1 \ldots c_n}(x_{c_1}, ..., x_{c_n}, y_{c_1}, ..., y_{c_n}) = \Big(1 + exp\Big(w_0 + w_1 d(m_0, m_1) + \frac{w_2}{|G_0|} \sum_{y_{c_j} \epsilon G_0} d(m_0, y_{c_j}) + \frac{w_3}{|G_1|} \sum_{y_{c_j} \epsilon G_1} d(m_1, y_{c_j})\Big)\Big)^{-1}$$

where ψ is a compatibility between the labels x_{c_i} and observations y_{c_i}. G_0 is one subgroup with a common label, having mean m_0 and G_1 is the other subgroup with a common label, having mean m_1. d is a distance measure between observations, which we define as a simple Euclidean distance.

[1] This is not defined for segments at the ends of a stroke.

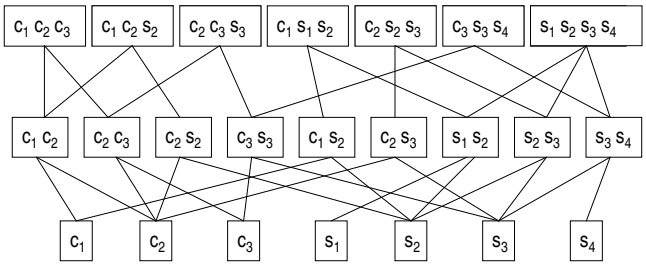


Figure 3: The region hierarchy for the graph shown in Figure 2. Messages are passed from a region to its subregions. Here each line connecting two rectangles represents a message.

The constants w_0, w_1, w_2,and w_3 are determined by hand. $\psi_{s_1 \ldots s_n}(x_{s_1}, ..., x_{s_n}, y_{s_1}, ..., y_{s_n})$ is defined similarly.

Note that this function is symmetric in that a labeling and the opposite labeling will have the same value, e.g. $\{corner, not\ corner, not\ corner\}$ will have the same value as $\{not\ corner, corner, corner\}$.

Functions of different node types

Finally, compatibility among adjacent nodes of different types is defined. These functions are specific to the types of features being considered. For detecting corners, two functions are defined, each with three arguments. The first takes one possible corner and two surrounding segments; the other takes one segment and two surrounding possible corners. In both cases the function is based on the intuition that a corner should have a larger direction change than the surrounding area, and that a corner should be relatively short (a large direction change over a very long distance generally does not look like a corner). This function is defined by:

$$\psi_{s_i c_i s_{i+1}}(x_{s_i}, x_{c_i}, x_{s_{i+1}} y_{c_i} y_{s_i} y_{s_{i+1}}) = (1 + exp(w_0 + w_1 f_1(s_i, c_i, s_{i+1}) + w_2 f_2(s_i, c_i, s_{i+1})))^{-1}$$

where f_1 is a function comparing direction change of each piece of the stroke and f_2 is a function comparing arc length. $\psi_{c_i s_i c_{i+1}}$ is defined similarly.

Determining a Labeling

Given a graph as described in the previous section, we want to find an assignment of labels to random variables which has maximum likelihood. Because a distribution over an undirected graph may be factored as a product of the potential functions of the maximal cliques, this likelihood can be written as:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(x_C) \prod_j \psi_j(x_j)$$

where Z is a normalization constant, x_C are all the nodes in clique C, and ψ is as defined previously. Dependence on the observations is omitted from the equation for simplicity.

Belief propagation is a commonly used method for solving approximate inference problems on graphs; such algorithms work by passing local messages until the messages

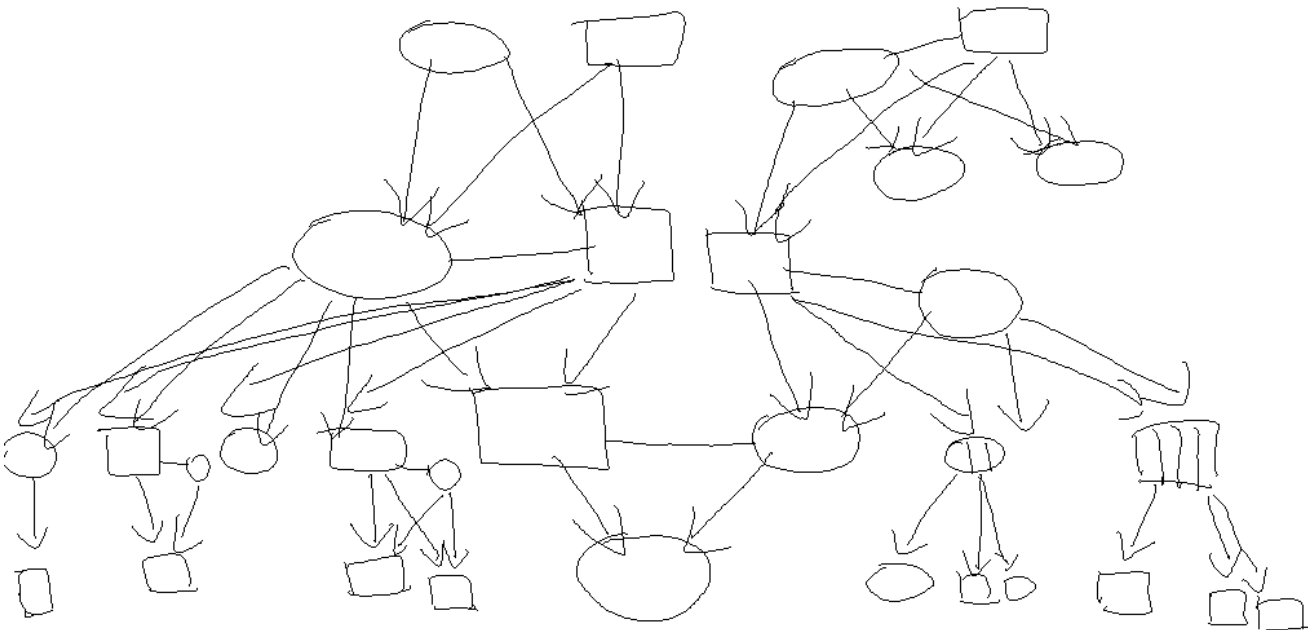


Figure 4: Example of a family tree sketch.

converge. Beliefs at each node (the certainty of a given label) are then computed as the product of the messages to the node and evidence at the node. A message from a to b maybe thought of as how certain a is that b has a particular labeling. Belief propagation is only well defined and guaranteed to converge for trees (acyclic graphs). However, it has been shown empirically that when applied to loopy graphs, belief propagation usually performs well (Murphy, Weiss, & Jordan 1999).

In ordinary belief propagation, messages are exchanged between single nodes. Yedidia, Freeman, and Weiss outline generalized belief propagation, in which messages pass between groups of nodes, called regions (2000), (2002). This is based on the intuition that these messages are more informative and will thus yield better results. Messages pass from regions to their direct subregions (which are defined by the intersection of regions). The belief of a region is based on evidence from within the region and messages going into that region from outside.

We apply generalized belief propagation to the inference problem formulated above. The maximal cliques in the graphs, described in a previous section, are selected as the regions. We omit functions among nodes that form non-maximal cliques for simplicity and because this information is already expressed by the larger functions we have defined.

Regions and subregions may be thought of as a hierarchy. Figure 3 shows the hierarchy of regions of the graph in Figure 2. Rectangles represent regions and subregions; messages are passed from a region to one of its subregions where two rectangles are connected.

Consider the region containing c_2 and c_3 and its subregion containing only c_2. For each, the belief for that region is the product of the evidence within the region and the messages entering the region (but not messages passed within the region). The beliefs of these two regions are expressed as:

$$\begin{aligned} b_{c_2c_3}(x_{c_2}x_{c_3}) &= \psi_{c_2}(x_{c_2})\psi_{c_3}(x_{c_3})m_{c_1\to c_2c_3}m_{s_3\to c_2c_3}\cdots \\ &\quad m_{c_1\to c_2}m_{s_2\to c_2}m_{s_3\to c_2}m_{s_3\to c_3} \end{aligned}$$

and

$$b_{c_2}(x_{c_2}) = \psi_{c_2}(x_{c_2})m_{c_1\to c_2}m_{c_3\to c_2}m_{s_2\to c_2}m_{s_3\to c_2}$$

where $m_{i\to j}$ represents a message from i to j. We may now find an expression for the message from c_3 to c_2, since

$$b_{c_2}(x_{c_2}) = \sum_{x_{c_3}} b_{c_2c_3}(x_{c_2}x_{c_3})$$

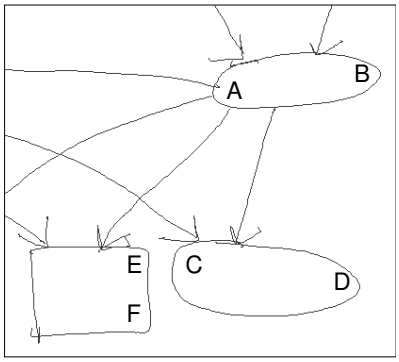
Here we have summed over all possible labels for c_3. Solving for $m_{c_3\to c_2}$ gives:

$$m_{c_3\to c_2} = \sum_{x_{c_3}} \psi_{c_3}(x_{c_3})m_{s_3\to c_2c_3}m_{c_1\to c_2c_3}m_{s_3\to c_3}$$

Expressions for all the messages may be found similarly. First messages are initialized and updated iteratively, then the beliefs for the individual nodes representing possible corners are computed. These beliefs represent the certainty of a given label ($corner$ or $not\ corner$) taking into account some context, indicated by edges in the graph.

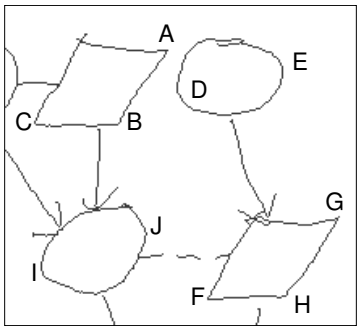
Results

We tested this approach to corner detection on sketches previously collected by Alvarado (2004). The data consists of hand drawn family trees in which females and males are represented by ellipses and rectangles and relationships (marriage, partnership, divorce, parent-child) are represented by



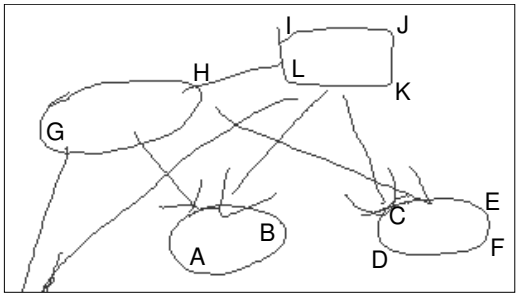
	1	2
A	.84	.48
B	.99	.98
C	.10	.01
D	.90	.36
E	.78	.88
F	.54	.66

a



	1	2
A	.98	.99
B	.25	.84
C	.99	.99
D	.21	.01
E	.63	.10
F	.99	.99
G	.99	.99
H	.98	.99
I	.60	.12
J	.26	.20

b



	1	2
A	.06	.01
B	.43	.06
C	.54	.42
D	.06	.05
E	.30	.30
F	.12	.10
G	.08	.00
H	.99	.91
I	.22	.75
J	.97	.99
K	.95	.99
L	.56	.91

c

Figure 5: Cropped areas of sketches from three different subjects. Tables to the right of each sketch give certainty values for each possible corner. The first column of numbers is before belief propagation; the second is after.

lines, jagged lines, and arrows respectively. This data is suitable for testing our system because it contains many actual corners (occurring in rectangles) and many curved areas that could be easily misinterpreted out of context (occurring most often in sides of the ellipses). The sketches, while often messy, are also easily interpreted by a human.

Figure 4 shows one of the sketches tested. Table 1 gives the results for this sketch, for the rectangles and ellipses only; we omit arrows and lines from these results because they are rarely ambiguous.

	without context	with context
correct possible corners (out of 55)	39 (71%)	47 (85%)
correct shapes (out of 31)	14 (45%)	21 (68%)

Table 1: Results from the sketch in Figure 4.

Possible corners are areas of the sketch with a direction change above a threshold; they are classified as either $corner$ or $not\ corner$. Fifty-five possible corners were found in this sketch of which 30 are actually corners. The results of the single node function described in a previous section are listed in the column "without context" for comparison. "Correct possible corners" indicates the number of possible corners that were given the correct label (this includes those that were correctly labeled as $not\ corner$). "Correct shapes" indicates the number of shapes (ellipses and rectangles) in which all of the possible corners were labeled correctly. Our results demonstrate that incorporating a small amount of context improves corner detection noticeably over a reasonable baseline.

Figure 5 contains three areas taken from larger sketches. These cropped areas show the output from our system more explicitly. Certainties without context (column 1) and with context (column 2) are listed in the tables beside each sketch. A possible corner with a certainty greater than .5 is regarded as a corner.

In the upper ellipse of Figure 5a, both possible corners (A and B) were assigned high certainty values by the single node compatibility function. Although both values were

lowered after belief propagation because of the influence of adjacent curved areas in the sketch, a shape with very strong but incorrect initial certainties can generally not be corrected. The lower ellipse has one possible corner with a high certainty (D), but the other (C) is identified correctly, leading to a correct labeling for D after belief propagation. The corners in the rectangle in this sketch were correctly identified initially; however, their certainties were raised by the belief propagation. In nearly all cases where all of the possible corners were labeled correctly initially, belief propagation strengthens the initial guess.

In Figure 5b, three of the four shapes have one possible corner that was initially misclassified. However, the correct labeling of the other possible corners and influences from the areas between the possible corners, which are clearly drawn as either straight or curved, fix the initial errors.

Figure 5c contains a case where belief propagation performed worse than the single node function. Possible corner I was not intended as a corner. We assume this partly because we prefer to interpret the shape as a rectangle rather than as a pentagon with one very short side. In this case, higher level domain knowledge is needed to correct the mistake.

Discussion and Future Work

The parts of sketches we use for identifying corners are possible corners and the segments in between them; however incorporating other aspects of a sketch could also contribute to finding corners. Also, we consider each stroke separately, but often several strokes are part of the same object. In these cases, influences between strokes would be useful, and a different criterion, based on distance, might be preferred for selecting the areas of the sketch over which to perform inference.

The problem of finding corners in a messy sketch motivated our approach, but there are other aspects of sketches that are also ambiguous, such as whether or not two strokes were intended to meet or whether two lines were intended to be parallel. We believe the approach we have presented could be generalized and applied to these other ambiguous situations.

Conclusion

This paper presents an approach to early sketch processing in which features such as corners are classified individually, but these features may be ambiguous when viewed out of context. We used a graphical model based approach to incorporate a limited amount of context in the classification. Results show an improvement when this approach is applied to the problem of detecting corners.

References

Alvarado, C. 2004. *Multi-Domain Sketch Understanding*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Coughlan, J. M., and Ferreira, S. J. 2002. Finding deformable shapes using loopy belief propagation. In *Proceedings of European Conference on Computer Vision*.

Gross, M. D., and Do, E. Y.-L. 2000. Drawing on the back of an envelope: a framework for interacting with application programs by freehand drawing. *Computers & Graphics* 24.

Hse, H.; Shilman, M.; and Newton, A. R. 2004. Robust sketched symbol fragmentation using templates. In *Proceedings of Intelligent User Interfaces*.

Murphy, K. P.; Weiss, Y.; and Jordan, M. I. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in Artificial Intelligence*.

Roth, D., and Yih, W. 2002. Probabilistic reasoning for entity & relation recognitions. In *Proceedings of International Conference on Computational Linguistics*.

Saund, E.; Mahoney, J.; Fleet, D.; Larner, D.; and Lank, E. 2002. Perceptual organization as a foundation for intelligent sketch editing. In *AAAI Spring Symposium on Sketch Understanding*.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2000. Generalized belief propogation. Technical Report TR-2000-26, Mitsubishi Electric Research Laboratory.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2002. Understanding belief propogation and its generalizations. Technical Report TR-2001-22, Mitsubishi Electric Research Laboratory.

Zhang, D.-Q., and Chang, S.-F. 2004. Learning to detect scene text using a higher-order mrf with belief propagation. In *IEEE Workshop on Learning in Computer Vision and Pattern Recognitions*.