Anatomical Sketch Understanding: Recognizing Explicit and Implicit Structure

Peter Haddawy[1], Matthew Dailey[2], Ploen Kaewruen[1], and Natapope Sarakhette[1]

[1] Asian Institite of Technology
[2] Sirindhorn International Institute of Technology, Thammasat University

Abstract. Sketching is ubiquitous in medicine. Physicians commonly use sketches as part of their note taking in patient records and to help convey diagnoses and treatments to patients. Medical students frequently use sketches to help them think through clinical problems in individual and group problem solving. Applications ranging from automated patient records to medical education software could benefit greatly from the richer and more natural interfaces that would be enabled by the ability to understand sketches. In this paper we take the first steps toward developing a system that can understand anatomical sketches. Understanding an anatomical sketch requires the ability to recognize what anatomical structure has been sketched and from what view (e.g. parietal view of the brain), as well as to identify the anatomical parts and their locations in the sketch (e.g. parts of the brain), even if they have not been explicitly drawn. We present novel algorithms for sketch recognition and for part identification. We evaluate the accuracy of the recognition algorithm on sketches obtained from medical students. We evaluate the part identification algorithm by comparing its results to the judgment of an experienced physician.

1 Introduction

Sketching is ubiquitous in medicine. Physicians commonly use sketches as part of their note taking in patient records and to help convey diagnoses and treatments to patients. Medical students frequently use sketches to help them think through clinical problems and to facilitate communication with other students when participating in group problem solving. Applications ranging from automated patient records to medical education software could benefit greatly from the richer and more natural interfaces that would be enabled by the ability to understand sketches. Our particular interest in sketch understanding stems from our work on the COMET collaborative intelligent tutoring system for medical problem-based learning (PBL) [12]. COMET provides a collaborative environment in which students from disparate locations can work together to solve clinical reasoning problems. It generates tutorial hints by using models of individual and group problem solving. The system provides a multi-modal interface that integrates text and graphics so as to provide a rich communication channel

between the students and the system, as well as among students in the group. While COMET has already proven itself useful [13], it still does not support the full range of interaction that occurs in human-tutored PBL sessions. In particular, it does not support interaction through sketches. From observation of PBL sessions at Thammasat University Medical School we have found that students typically sketch anatomical structures on the white board while solving a problem. The sketches are used to help think through the problem and as an artifact to support communication among the students. Consider the following scenario:

> A group of students in a PBL session is given a problem concerning unconsciousness due to a car accident. One student sketches the brain. Thinking about direct impact to the head, another student annotates the sketch to indicate a contusion in the area where the frontal lobe should be, although the frontal lobe was not explicitly drawn. The tutor understands this annotation and encourages the students to also consider damage to the brain stem by pointing to that part of the sketch and saying "think about what is going on here as well".

Supporting this kind of interaction requires several capabilities. First is the ability to recognize what anatomical structure or structures have been sketched and from what perspective (e.g. parietal view of the brain). Next is the ability to identify anatomical parts of the sketched structure (e.g. frontal lobe of the brain), even if they have not been explicitly drawn. Finally is the ability to understand annotations on the sketch and to be able to effectively use the sketch as a medium of communication in a dialogue. In this paper we address the first two issues. We present a novel approach to sketch recognition that combines the use of shape context matching [3] together with continuous Naive Bayes classification. The approach is robust and is insensitive to scaling. Next we present an algorithm that uses shape context matching in yet another way to identify the parts of the anatomical structure. The algorithm works even if the proportions in the sketch are not anatomically correct and whether or not the anatomical parts have been explicitly drawn. We evaluate the sketch recognition algorithm on a collection of sketches by medical students of various views of the brain, heart, and lungs. Our algorithm achieves a recognition accuracy of 73.6%, far above the baseline random classification accuracy of 12.5%. We evaluate the part identification algorithm by comparing its results to those of an experienced physician. Location, orientation, size, and shape of the parts identified by the physician and the algorithm are in close agreement.

2 Related Work

The last few years has seen a tremendous increase in interest in sketch-based interfaces. Applications include computer-aided design, knowledge acquisition, and image retrieval. Researchers in this area emphasize that the informalness of sketches is important because it communicates that fact that the ideas being represented are still rough and thus invites collaboration and modification.

Clean, precise-looking diagrams created by most graphics programs can produce an impression of more precision than was intended and can lead to a feeling of commitment to a sketch as originally drawn [9, 7]. We now discuss a few systems that are representative of the state-of-the-art.

The Electronic Cocktail Napkin [7] is a general-purpose sketching program that provides trainable symbol recognition, parses configurations of symbols and spatial relations, and can match similar figures. It recognizes a symbol by comparing its features — pen path, number of strokes and corners, and aspect ratio — with a library of stored feature templates. Applications developed using the system include a visual bookmark system, an interface to simulation programs, and an HTML layout design tool.

SILK [10] is a sketching tool for developing user interfaces. SILK recognizes seven basic widgets, as well as combinations of widgets. To recognize a widget, SILK first identifies primitive components using a statistical classifier learned from examples. SILK recognizes four single-stroke primitive components: rectangle, squiggly line, straight line, and ellipse. Once components are identified, they are passed to an algorithm that detects spatial relationships among primitive and widget components. These include containment, closeness, and sequence. SILK finally uses a set of rules to identify widgets from primitive components. In an evaluation with twelve users, SILK achieved a widget recognition accuracy of 69%. SILK supports use of five single-stroke gestures for editing sketches: cross, circle, squiggly line, spiral, and angle (for insertion). Designers can create storyboards by drawing arrows from any screen's graphical objects, widgets, or background to another screen. SILK has a run mode in which it can simulate the functioning of the widgets and the transitions between screens.

ASSIST [2] supports sketching and simulation of simple 2-dimensional mechanical systems. ASSIST recognizes the user's sketch by identifying patterns that represent mechanical parts, leveraging off the fact that mechanical engineering has a fairly concrete visual vocabulary for representing components. ASSIST uses a three-stage procedure to choose the most likely interpretation for each stroke. First it matches the stroke to a set of templates to produce the set of possible interpretations, e.g. circle or rectangle. Next it ranks the interpretation using heuristics about drawing style and mechanical engineering. Finally, the system chooses the best consistent overall set of interpretations and displays this to the user. ASSIST supports editing of the sketch through the use of gestures. At any time during the design process, the user can run a simulation of the design being sketched.

In an effort to attain immediate practical functionality as well as broad domain independence, Forbus and Usher [5] take a very different approach to sketching. Their sKEA system does not address the recognition issue, focusing rather on qualitative reasoning about the spatial relations among objects and on analogical comparison of sketches containing multiple objects. They avoid the recognition problem by requiring the user to indicate when he begins and finishes drawing a new object as well as the interpretation of the object. The interpretation is selected from a pull-down menu.

The work reported in this paper is the first application of sketch-based interfaces to intelligent tutoring that we know of, and also the first in a medical domain other than image retrieval [1]. The motivation behind the use of sketching in medical tutoring is similar to that previously mentioned, namely that sketching supports collaboration and encourages modification. But in addition, sketching in medical PBL is valuable because it gives students practice in recalling anatomical structure. A menu-based drawing interface would not provide such practice. The issues involved in recognizing anatomical sketches are significantly different from those of recognizing design diagrams. Most of the previous work in sketching starts by recognizing primitive components such as lines, circles, and corners. This works fine for domains such as mechanical engineering and user interface design, but anatomical sketches are rather amorphous complex structures which may be sketched with more or less detail. This complexity and lack of a well-defined set of primitive components demands a very different approach to object recognition. Fortunately, the anatomical recognition problem is eased by the fact that by convention 2-dimensional depictions of anatomical structures are only shown from eight standard views. We have five external views corresponding to the sides of a cube: anterior, posterior, superior, inferior, lateral (2 sides); and three internal views corresponding to the three cutting planes: sagittal, coronal, axial. This fact is exploited by our recognition algorithm, described next.

3 Recognizing Structure and Parts

We call our prototype system UNAS[3] for UNderstanding Anatomical Sketches. We divide the task of understanding a sketch into two subtasks: identifying *what* the sketch portrays, then identifying the relevant *parts* of the sketch.

Without constraints, this problem would be extremely difficult, if not impossible. Fortunately, the fact that 2-dimensional anatomical sketches are always drawn from one of eight standard views allows us to cast the problem of identifying what a sketch portrays as a *classification* problem: given an image of a sketch $\mathcal{I}$, find the class $y = f(\mathcal{I}) \in \{1, \ldots, K\}$ to which the image belongs. The set of possible classes corresponds to the set of standard views of anatomical structures, e.g., "parietal view of the brain" and "internal view of the lungs." With enough labeled examples $\{(\mathcal{I}_1, y_1), \ldots, (\mathcal{I}_m, y_m)\}$, it is possible to construct a classifier $\hat{y} = h(\mathcal{I})$ that predicts the unknown true class $y = f(\mathcal{I})$ given a previously unseen $\mathcal{I}$.

Once we assume the class y that sketch $\mathcal{I}$ belongs to, we must then *segment* the sketch into regions corresponding to anatomical parts. Since every instance of a standard anatomical view contains the same parts, the task is well-defined: attach a label $z \in \{1, \ldots, L_y\}$ to every pixel in $\mathcal{I}$. Here the set of possible labels corresponds to the set of anatomical parts normally visible in view y, e.g., "temporal lobe" and "parietal lobe."

[3] Unas was the last king of the 5th dynasty of ancient Egypt. The interpretation of the bas-relief scenes on the inside of his tomb remains a challenge to this day.

In the preliminary experiments reported upon in this paper, we have made the following simplifying assumptions:

- Each image $\mathcal{I}$ contains exactly one anatomical structure, e.g. brain, lungs, heart.
- Sketches may not contain annotations or extraneous parts.
- Each sketch is complete (there are no major parts left out).

In future work, we plan to relax all of these assumptions.

For the classification problem, we take the Bayesian maximum a posteriori (MAP) approach: measure a finite set of features $x_1, \ldots, x_n$ from $\mathcal{I}$ then select the class

$$\hat{y} = \arg\max_y P(y \mid \mathbf{x})$$

where

$$P(y \mid \mathbf{x}) \propto P(\mathbf{x} \mid y)P(y).$$

$P(\mathbf{x} \mid y)$ is the likelihood of feature vector $\mathbf{x}$ given class y, and $P(y)$ is the prior probability of class y. We estimate the parameters of statistical model $P(\mathbf{x} \mid y)$ from training data, and in the experiments reported in this paper, we assume uniform priors $P(y)$. In some contexts, such as a PBL session, however, the priors could be chosen to reflect our prior knowledge that, for example, in a head injury case study, sketches of the brain are more likely than sketches of the lungs.

The MAP classifier just described requires a set of features and a model for the data likelihood. In our scheme, feature x_i for sketch $\mathcal{I}$ is the dissimilarity between $\mathcal{I}$ and template image $\mathcal{T}_i$ according to Belongie et al.'s Shape Context measure [3]. Ideally, the set of templates $\mathcal{T}_i$ contains several examples of each class. Our model for the data likelihood is the well-known Naive Bayes model

$$P(\mathbf{x} \mid y) = \prod_i P(x_i \mid y).$$

The model is "naive" in that it assumes the feature values x_i are statistically conditionally independent given y, even though they generally are not.

Once our Naive Bayes classifier picks the best class $\hat{y}$ for a given input sketch $\mathcal{I}$, the next step is to segment the sketch into regions. Our system first warps the input sketch into correspondence with a pre-labeled *canonical* template $\mathcal{T}^*_{\hat{y}}$ for class $\hat{y}$, assigns labels to sketch points using the labels in $\mathcal{T}^*_{\hat{y}}$, finds the boundary of each region, then labels each pixel in the sketch according to which region it falls into.

We compute point correspondences between $\mathcal{I}$ and $\mathcal{T}^*_{\hat{y}}$ using (once again) Belongie et al.'s Shape Context algorithm [3]. We then use the point correspondences to estimate a mapping between arbitrary points in the sketch and template using the Thin Plate Spline (TPS) model [4]. To identify the boundary of each region, we transfer the labeled points from $\mathcal{T}^*_{\hat{y}}$ to $\mathcal{I}$ then connect those points using a simple traveling salesperson algorithm [8].

In the rest of this section, we describe the sketch classification and segmentation algorithms in more detail.

3.1 Sketch Classification

As previously described, the basic features in our Naive Bayes classifier are dissimilarities between the input sketch image $\mathcal{I}$ and each of a set of template images $\mathcal{T}_i$. The particular dissimilarity measure we use is Belongie et al.'s Shape Context (SC) measure [3]. SC represents a shape as a set of points sampled from the shape's contours. Each sample point is represented by a coarse histogram of the other points surrounding it. To determine the dissimilarity of two shapes, SC first finds a correspondence between the sampled points in the two shapes. Then the total dissimilarity between the shapes is simply the sum of the dissimilarities of the sample points.

For each template image, we convert the raw grayscale or color image to a line drawing then In either case, we randomly sample N_s points from the resulting "edge" image. For each point p_i, we obtain the SC histogram by counting the number of pixels falling into N_b log-polar bins around p_i then normalizing the bin counts (so the sum of the bin counts is 1). The width of the bin template is adjusted to be proportional to the mean squared distance between points, to make the resulting histograms invariant to the scale of the image.

For a new sketch, we perform the same sampling and SC histogram computation steps, then find the optimal correspondence between the sketch sample points and the template's sample points. The dissimilarity between two normalized SC descriptors is the simply the χ^2 test statistic. Given the (square) dissimilarity matrix for the sketch and template SC histograms, the optimal correspondence is the permutation of the sketch points minimizing the summed dissimilarity of the matched points. This corresponds to a weighted bipartite graph matching problem and is solved in $O(N_s^3)$ time using the Hungarian method [11]. Once we obtain the optimal assignment, the final dissimilarity x_i between sketch $\mathcal{I}$ and template $\mathcal{T}_i$ is the sum of the N_s individual point-matching costs.

After computing the dissimilarities x_i between sketch $\mathcal{I}$ and templates $\mathcal{T}_i$, UNAS forms the feature vector $\mathbf{x} = [x_1, \ldots, x_n]^T$, which is then input to the sketch classifier.

UNAS assumes each of the probability densities $P(x_i \mid y)$ used in the Naive Bayes classifier is a Gaussian with mean $\mu_{y,i}$ and standard deviation $\sigma_{y,i}$. The classifier's $2nK$ parameters $\mu_{y,i}, \sigma_{y,i}$ are estimated directly from a training set containing an equal number of example sketches from each class.

Once UNAS obtains the MAP estimate $\hat{y}$ for the class of $\mathcal{I}$, the next step is to segment the sketch into regions corresponding to anatomical parts. We describe the details of the segmentation procedure next.

3.2 Sketch Segmentation

As previously described, the first step in segmentation is to align sketch $\mathcal{I}$ with the canonical labeled template $\mathcal{T}^*_{\hat{y}}$ for class $\hat{y}$. To align the sketch with the template, UNAS first uses Shape Context as described above to find a set of N_s point correspondences $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$ between the sketch and the template. These correspondences are then used to fit a thin plate spline (TPS) model [4]

mapping $\mathcal{T}_{\hat{y}}^*$ to $\mathcal{I}$. TPS fits a smooth function $f_x(x, y)$ mapping the template points (x_i, y_i) to the x coordinates x'_i of the sketch points, and another smooth function $f_y(x, y)$ mapping the template points to the y coordinates y'_i of the sketch points. The fitted functions f_x and f_y model the deformation of thin steel plates constrained to interpolate the observed values x'_i and y'_i, respectively. However, since sampling introduces noise, and the Hungarian assignment method does not attempt to impose any spatial regularity constraints, strict interpolation is not desirable. Belongie et al. [3] introduce a regularization factor into the minimization that penalizes excessively warped transformations. The quality of the final transform can be iteratively improved by repeating the correspondence estimation and transform estimation steps, using the results of the previous step as a starting point. In our experiments, we iterate the process 6 times. The result is a smooth mapping from every point in $\mathcal{T}_{\hat{y}}^*$ to a point in $\mathcal{I}$.

The canonical templates $\mathcal{T}_y^*$ are derived from drawings in medical atlases, for which the ground truth segmentation is known. When we sample and compute the SC histograms for each canonical template, we also associate (by hand) a set of labels with each sampled point. The labels indicate which regions (anatomical parts) each point belongs to. Since the sampled points correspond to edges in the original image, they often delineate boundaries between two regions; in these cases, the points are assigned the labels of both regions.

We initiate the segmentation process by simply copying the labels of the template points (x_i, y_i) to the corresponding points $(f_x(x_i, y_i), f_y(x_i, y_i))$ in $\mathcal{I}$. Now the task is to use these points to compute a closed boundary for each region of $\mathcal{I}$. Under certain conditions described by Giesen [6], solutions to the *traveling salesperson tour problem* (TST) accomplish exactly this task.

We use Giesen's insight for curve reconstruction in UNAS. For each anatomical part label $z_i \in \{1, \ldots, L_{\hat{y}}\}$ for view $\hat{y}$, UNAS collects the set of projected boundary points for region z_i and runs a traveling salesperson algorithm [8] to "connect the dots." The result is a simple polygon approximating the boundary of region z_i in $\mathcal{I}$.

The final step, after the region boundaries have been determined, is to use those boundaries to assign a unique label z to each pixel of $\mathcal{I}$. UNAS tests each pixel for membership in each polygonal region using the technique of segment intersections: if an arbitrary ray from pixel p intersects an odd number of the polygon's sides, it is inside the polygon; otherwise, it is outside the polygon.

This concludes our description of the classification and segmentation algorithms employed by UNAS. In the next section, we describe an empirical evaluation of the approach.

4 Evaluation

We evaluated the sketch classification algorithm by building a Naive Bayes classifier for the brain, heart, and lungs and evaluating its accuracy in classifying sketches. We chose the following eight views:

- Brain: parietal (lateral), sagittal, basal (inferior)

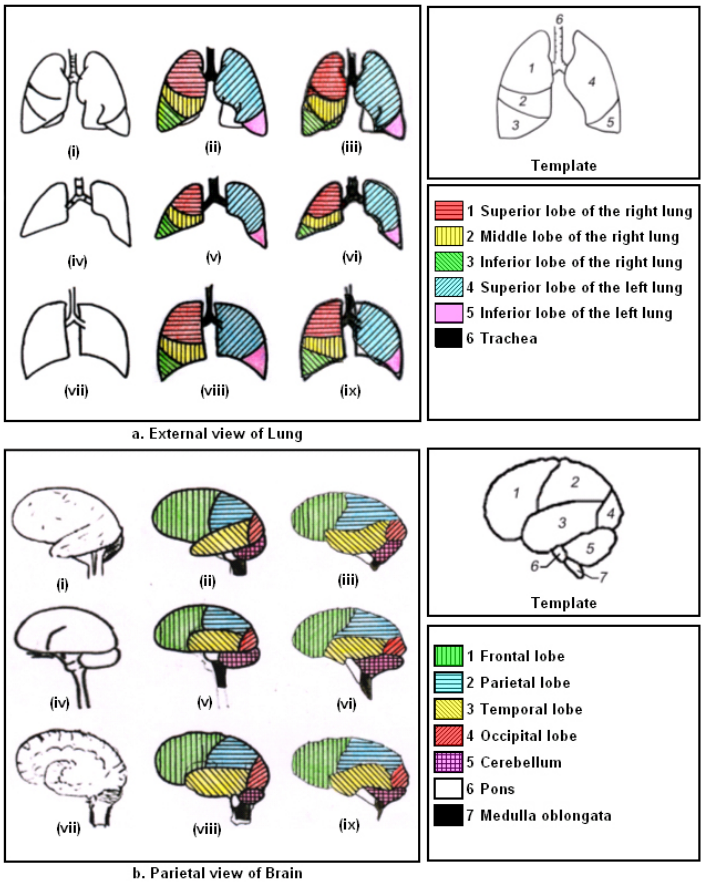

Fig. 1. Medical student sketches (first column) with corresponding segmentations produced by a physician (second column) and by UNAS (third column). The templates used by the segmentation algorithm are shown in the upper right corners.

- Heart: anterior, posterior, interior (coronal)
- Lung: anterior, interior (coronal)

These views were chosen because they are the standard views from which these organs are typically drawn. We collected 300 sketches from 48 medical students in their second to sixth years of study. The sketches were vetted for quality by a physician and we eliminated those that the physician could not identify. This was done because we do not expect our recognition algorithm to perform better than an experienced physician and because low quality sketches are unlikely to be useful as templates. This left us with 272 sketches. We then chose an equal number of sketches for each view, resulting in 30 sketches for each view or a total of 240 sketches. For each view we randomly separated the sketches into 70% (21

sketches) for training and 30% (9 sketches) for testing. All the 168 sketches in the training set were potential templates in the Naive Bayes classifier. To this set we added six medical atlas illustrations for each view, resulting in a total of 216 candidate templates. The 168 sketches in the training set were used to compute the means $\mu_{y,i}$ and variances $\sigma_{y,i}$ of eight conditional Gaussian distributions $P(x_i \mid y)$ for each candidate template $\mathcal{T}_i$. Using all the templates would result in a classifier with unacceptably slow running time and also may not yield the highest classification accuracy. So we conducted feature selection by performing a best-first search through the space of all subsets of templates. Each subset resulted in a different Naive Bayes classifier, that was evaluated on a validation set, using 7-fold cross validation (to evenly divide the training set of 21 sketches per view). The best performing classifier contained 24 templates. This classifier had a total classification accuracy of 73.6% on the test set, far above the baseline random classification accuracy of 12.5%. The accuracy for each class ranged from the lowest value of 55.6% for the brain parietal and heart posterior views to the highest of 88.9% for the heart anterior and lung external views.

We evaluated the sketch segmentation algorithm by comparing its segmentation to that of an experienced physician on three sketches each of the external view of the lungs and the parietal view of the brain. We chose three qualitatively different sketches for each organ. Each sketch used was correctly recognized by UNAS. The segmentation results are shown in Figure 1. The first column shows the sketches, the second column is the physician's segmentation, and the last column is UNAS's segmentation. The segmentations produced by UNAS and by the physician agree quite closely on all sketches. For example, in the first sketch of the lung (i) the student drew a protrusion below the superior lobe of the left lung that is not normally drawn in the external view. Both the physician and UNAS correctly did not include this as part of the superior lobe. The segmentations produced by UNAS differ from of those of the physician in two primary respects. When internal parts are drawn slightly incorrectly, the physician still segments following the lines in the sketch. In contrast, UNAS attempts to correct the sketch. This can be seen by comparing the superior lobe of the left lung in viii and ix, and the cerebellum in viii and ix. The other difference is that because the thin-plate spline transformation is applied globally, sometimes parts get warped too much, for example the pons in segmentation vi.

5 Conclusions and Future Research

The results from our initial prototype system are encouraging but much work remains to be done in order to realize the functionality described in our motivating example. We are currently gathering sample sketches of more anatomical structures to expand the scope of UNAS. We also plan to compare the recognition accuracy and segmentation results of UNAS to those of physicians with varying levels of experience. On the algorithm side, several improvements and extensions can be made. The accuracy of the recognition algorithm can be improved. A first step is to use a more sophisticated search for feature selection

since the feature space seems to contain many local maxima. A next step is to relax the Gaussian assumption in the Naive Bayes model, but this will require more examples. Other machine learning techniques that direclty use similarity information, such as support vector machines with similarity kernels, are promising and should be tried. We have assumed that a sketch includes only one anatomical structure but sketches often contain multiple structures as well as incompletely drawn structures. Generalizing our approach to handle this will possibly require adding spatial reasoning abilities. In addition to understanding the sketch, UNAS should be able to understand annotations commonly used in medicine, such as arrows, circles, crosses, darkened regions, and clusters of dots. For this we are exploring the use of hidden Markov models, which tend to work well for such relatively simple symbols. The final step will be to integrate UNAS into the COMET intelligent tutoring system.

References

1. A. F. Abate, M. Nappi, G. Tortora, and M. Tucci. Assisted browsing in a diagnostic image database. In *AVI '96: Proceedings of the workshop on Advanced visual interfaces*, pages 223–232, 1996.
2. C. Alvarado and R. Davis. Resolving ambiguities to create a natural sketch-based interface. In *Proc. International Joint Conf. on Artificial Intelligence*, August 2001.
3. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.
4. F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.
5. K. Forbus and J. Usher. Sketching for knowledge capture: A progress report. In *Proc. International Conference on Intelligent User Interfaces*, pages 71–77, 2002.
6. J. Giesen. Curve reconstruction in arbitrary dimension and the traveling salesman problem. In *Proc. 8th Discrete Geometry and Computational Imagery (DGCI) Conference*, pages 164–176, 1999.
7. M. Gross. The proverbial back of an envelope. *IEEE Intelligent Systems*, pages 10–13, May/June 1998.
8. N. Gulley. Traveling salesman problem demonstration (MATLAB code), 1993.
9. M. Hearst. Sketching intelligent systems. *IEEE Intelligent Systems*, page 10, May/June 1998.
10. J. Landay and B. Myers. Sketching interfaces: Toward more human interface design. *IEEE Computer*, pages 56–64, March 2001.
11. C. Papadimitriou and K. Stieglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1982.
12. S. Suebnukarn and P. Haddawy. A collaborative intelligent tutoring system for medical problem-based learning. In *Proc. International Conference on Intelligent User Interfaces*, pages 14–21, 2003.
13. S. Suebnukarn and P. Haddawy. Clinical-reasoning skill acquisition through intelligent group tutoring. In *Proc. International Joint Conf. on Artificial Intelligence*, August 2005.