Artificial Intelligence in Medicine (2006) xxx, xxx–xxx

ELSEVIER

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Anatomical sketch understanding: Recognizing explicit and implicit structure☆

Peter Haddawy*, Matthew N. Dailey, Ploen Kaewruen, Natapope Sarakhette, Le Hong Hai

Computer Science and Information Management Program,
School of Engineering and Technology,
Asian Institute of Technology, PO Box 4,
Klong Luang, Pathumthani 12120, Thailand

Received 16 January 2006; received in revised form 15 July 2006; accepted 18 July 2006

KEYWORDS
Anatomical sketches;
Problem-based learning;
Human–computer interaction;
Object recognition;
Image segmentation

Summary

Objective: 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.
Methods: 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.
Results: The sketch recognition algorithm achieves a recognition accuracy of 75.5%, far above the baseline random classification accuracy of 6.7%. Comparison of the results of the part identification algorithm with the judgment of an experienced

☆ This is a revised and extended version of the paper, ''Anatomical sketch under standing: Recognizing explicit and implicit structure,'' by P. Haddawy, M.N. Dailey, P. Kaewruen, and N. Sarakhette, presented in the 10th Conference on Artificial Intelligence in Medicine (AIME 05).

* Corresponding author. Tel.: +66 2 524 5002; fax: +66 2 524 5003.
E-mail addresses: haddawy@ait.ac.th (P. Haddawy), mdailey@ait.ac.th (M.N. Dailey), Ploen.Kaewruen@ait.ac.th (P. Kaewruen), Natapope_Sa@truecorp.co.th (N. Sarakhette), LeHong.Hai@ait.ac.th (L.H. Hai).

0933-3657/$ – see front matter
doi:10.1016/j.artmed.2006.07.010

physician shows close agreement in terms of location, orientation, size, and shape of the identified parts.
Conclusions: The performance of our prototype in terms of accuracy and running time provides strong evidence that development of robust sketch understanding systems for medical domains is an attainable goal. Further work needs to be done to extend the approach to sketches containing multiple and partial anatomical structures, as well as to be able to interpret sketch annotations.

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) [1,2]. 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 [3], 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 dialog. 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 [4] together with support vector machine (SVM) 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 75.5%, far above the baseline random classification accuracy of 6.7%. 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

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

acquisition, and image retrieval. Researchers in this area emphasize that the informality 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 [5,6]. We now discuss a few systems that are representative of the state of the art.

The Electronic Cocktail Napkin [6] 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 [7] 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 [8] supports sketching and simulation of simple two-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 [9] 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 [10]. 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 two-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, and lateral (two sides); three internal views corresponding to the three cutting planes: sagittal, coronal, and axial. This fact is exploited by our recognition algorithm, described next.

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

3. Methods for recognizing structure and parts

We call our prototype system UNAS[1] 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 two-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:

$$y = f(\mathcal{I}) \in \{1, \ldots, N\}$$

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 *template* drawings or sketches:

$$\{\langle \mathcal{T}_1, f(\mathcal{T}_1)\rangle, \ldots, \langle \mathcal{T}_m, f(\mathcal{T}_m)\rangle\},$$

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 $\mathcal{I}$.

In the preliminary experiments reported upon in this paper, we have made the following simplifying assumptions about input sketches, examples of which are shown in Fig. 1:

- Each image $\mathcal{I}$ contains exactly one anatomical structure, e.g. brain, skull, lungs, heart, stomach, or kidneys.
- 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, UNAS combines template matching with machine learning. We begin with the labeled templates:

$$\{\langle \mathcal{T}_1, f(\mathcal{T}_1)\rangle, \ldots, \langle \mathcal{T}_m, f(\mathcal{T}_m)\rangle\}$$

including examples of each anatomical view. When the user creates a sketch $\mathcal{I}$, UNAS compares $\mathcal{I}$ to each of the stored templates using Belongie et al.'s shape context (SC) distance (dissimilarity) measure $d^{SC}(\cdot,\cdot)$ [4].

A typical template matching approach to predict the category $y = f(\mathcal{I})$ of the sketch would be to select the category of the template rated most similar to I:

$$\hat{y} = h(\mathcal{I}) = f(\arg\min_{\mathcal{T}_i} d^{SC}(\mathcal{I}, \mathcal{T}_i)).$$

However, we find that better classification performance can be obtained with the aid of machine learning. Our current prototype employs a support vector machine (SVM) classifier. Our SVM is a function $h : \mathbb{R}^m \mapsto \{1, \ldots, N\}$ mapping a feature vector:

$$\boldsymbol{x}_{\mathcal{I}} = [d^{SC}(\mathcal{I}, \mathcal{T}_1), \ldots, d^{SC}(\mathcal{I}, \mathcal{T}_m)]^T$$

of dissimilarities to a category $\hat{y}$. The classifier $h(\cdot)$ is learned from a training set of labeled feature vectors:

$$\{\langle \boldsymbol{x}_{\mathcal{I}_1}, f(\mathcal{I}_1)\rangle, \ldots, \langle \boldsymbol{x}_{\mathcal{I}_n}, f(\mathcal{I}_n)\rangle\}.$$

Once our SVM classifier picks the best class 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 prelabeled *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 [4]. 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 [11]. 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 [12].

3.1. Sketch classification

As explained above, the basic features in our SVM classifier are dissimilarities between the input sketch $\mathcal{I}$ and each of a set of templates $\mathcal{T}_i$. The particular dissimilarity measure we use is Belongie et al.'s shape context (SC) measure [4]. 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 and then computes an aligning transformation that maps the first shape onto the other. Once the aligning transformation is found, the total dissimilarity between the shapes is simply the sum of the dissimilarities of the sample points, plus the cost of the alignment of the matching points.

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

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

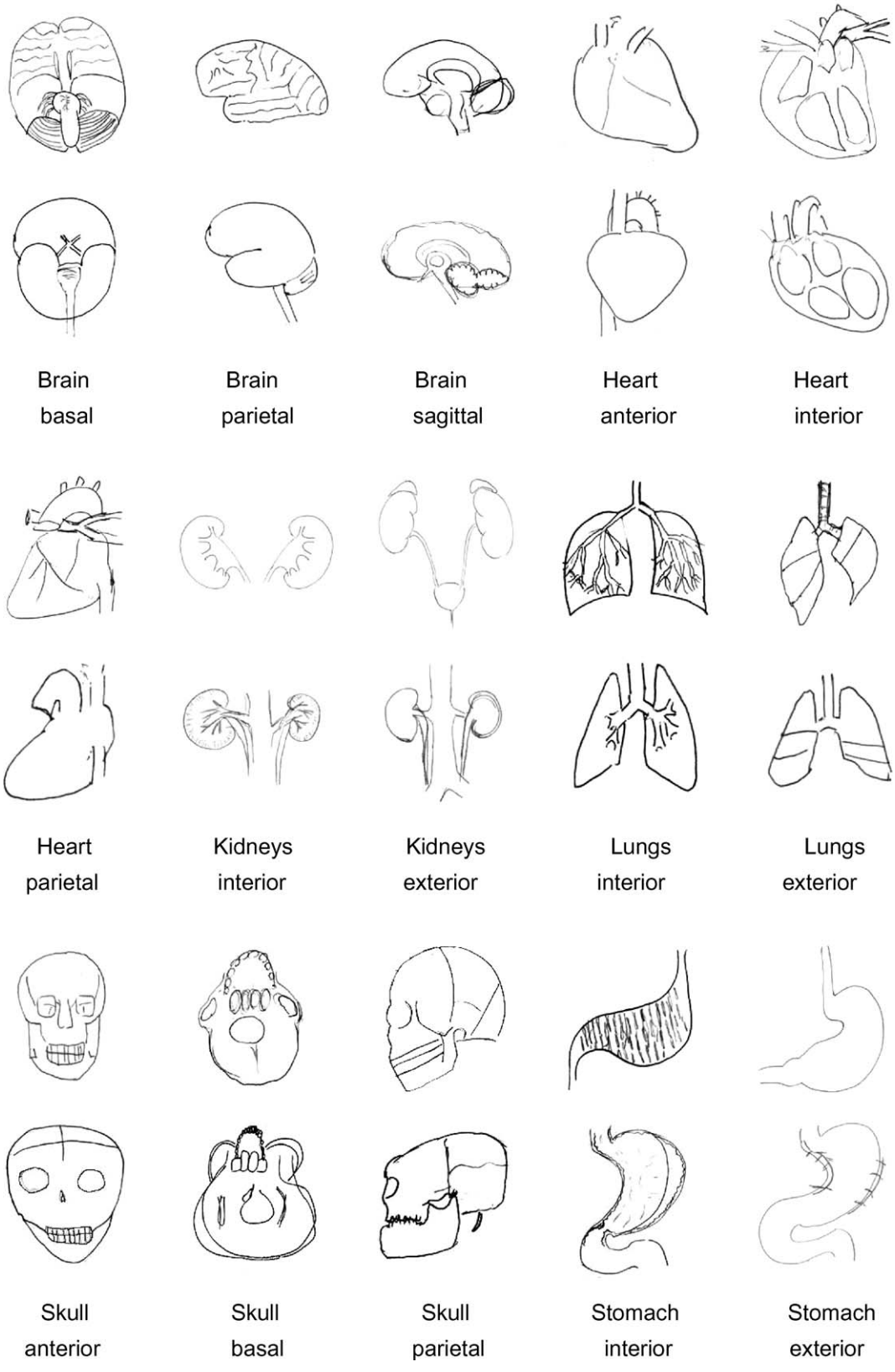

Figure 1 Sample anatomical sketches correctly recognized by UNAS.

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

We now describe each of these steps and the SVM classifier in detail. Template processing is performed as a preprocessing step offline; new sketches are processed and classified online.

3.1.1. Template preprocessing

The first step of template processing is to convert each image into a bitmap, using one of two methods:

- If the template is a simple line drawing, we convert it to a bitmap using Otsu's method [13], which chooses a threshold minimizing the variance of the pixel intensities above and below the threshold. We then perform line thinning on the resulting binary image to obtain a uniform line width throughout the image.
- If the template is a textured or shaded image, we perform Sobel edge detection with suppression of weak and nonmaximum points.

Fig. 2 shows the template sketches used in our current classification prototype. Some are derived from drawings in medical atlases, and others were drawn by medical students at Thammasat University.

Referring to the sketches in Fig. 1 and the templates in Fig. 2, the reader will notice that whereas the *internal details* of medical sketches might vary for a given view, the *outline* is relatively stable. One dramatic example is the two interior views of the lungs in Fig. 1. This variability makes it difficult to obtain accurate point correspondences for internal features, but easier to obtain accurate point correspondences along sketch outlines. To exploit this situation, we manually segment each template $\mathcal{T}_i$ into an *outline* region $\mathcal{T}_i^{\text{outl}}$ and an *interior* region $\mathcal{T}_i^{\text{int}}$, as shown in Fig. 3(a)–(c). Only the outline region is used to establish point correspondences.

After thinning and segmenting each template $\mathcal{T}_i$ into regions $\mathcal{T}_i^{\text{outl}}$ and $\mathcal{T}_i^{\text{int}}$, we obtain uniform

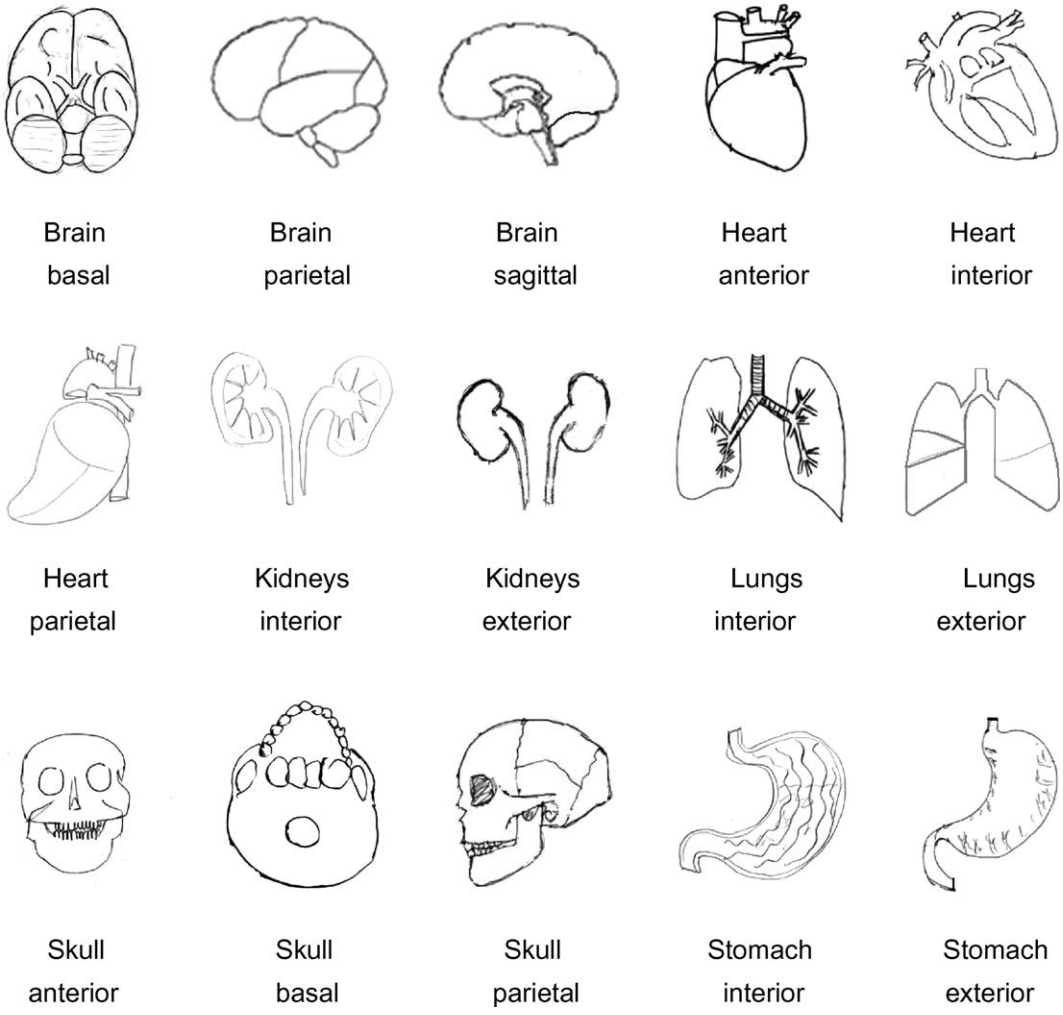

Figure 2 Templates used for sketch classification in the current UNAS prototype.

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

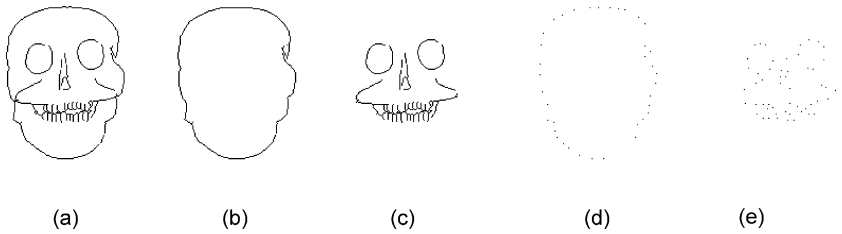

Figure 3 Template point sampling example: (a) original thinned template, (b) outline region, (c) interior region, (d) outline sample, and (e) interior sample.

random samples $\mathcal{P}_{\mathcal{T}_i}^{\text{outl}}$ and $\mathcal{P}_{\mathcal{T}_i}^{\text{int}}$ from the points in each region. The sample sizes are proportional to the sizes of the original sets, i.e.:

$$|\mathcal{P}_{\mathcal{T}_i}^{\text{outl}}| = \lfloor c_s |\mathcal{T}_i^{\text{outl}}| \rfloor \quad \text{and} \quad |\mathcal{P}_{\mathcal{T}_i}^{\text{int}}| = \lfloor |c_s \mathcal{T}_i^{\text{int}}| \rfloor .$$

For the results reported in this paper, we use $c_s = 1/11$. Example outline and interior samples are shown in Fig. 3(d) and (e).

Once we obtain the sampled point sets $\mathcal{P}_{\mathcal{T}_i}^{\text{outl}}$ and $\mathcal{P}_{\mathcal{T}_i}^{\text{int}}$ from template $\mathcal{T}_i$, for each point $p_j \in \mathcal{P}_{\mathcal{T}_i}^{\text{outl}}$, we compute two shape context histograms. The first histogram, used for point correspondence estimation, only includes information about the *outline* of the template. The outline-only histogram makes point matching robust to the amount of detail in the interior of the sketch. The second histogram, used for the final assessment of similarity between the template and sketch, also includes information about the interior of the template. This makes the total similarity assessment sensitive to the details in the interior of the template, so that given two templates with similar outlines, the best match for a sketch will be the one whose interior details are most similar to those of the sketch.

To obtain the outline-only histogram, $\boldsymbol{h}^{\text{SC}}(p_j, \mathcal{P}_{\mathcal{T}_i}^{\text{outl}})$, used for point correspondence estimation, we count the number of pixels within $\mathcal{P}_{\mathcal{T}_i}^{\text{outl}}$ falling into each of a set of log-polar bins around p_j and normalize 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. See Fig. 4 for an example. The second histogram

$$\boldsymbol{h}^{\text{SC}}(p_j, \mathcal{P}_{\mathcal{T}_i}^{\text{outl}} \cup \mathcal{P}_{\mathcal{T}_i}^{\text{int}}),$$

used to calculate total dissimilarity between the sketch and $\mathcal{T}_i$, is calculated the same way but incorporates information about both outline and interior points.

3.1.2. Online sketch processing

When a UNAS user draws a new sketch $\mathcal{I}$, we must obtain shape context histograms from $\mathcal{I}$ for comparison with the templates. Our prototype does not yet have the ability to automatically segment a sketch into outline and interior regions, so we simply obtain a single sample $\mathcal{P}_i$ from the thinned version of $\mathcal{I}$. The sample size is the same as for the templates, i.e.:

$$|\mathcal{P}_{\mathcal{I}}| = \lfloor c_s |\mathcal{I}^{\text{thin}}| \rfloor$$

For each point $p_j \in \mathsf{P}_{\mathsf{I}}$, we compute the shape context histogram $\boldsymbol{h}^{\text{SC}}(p_j, \mathcal{P}_i)$ using all of the sampled points.

After computing the shape context histograms for the sampled sketch points, we assess the similarity of $\mathcal{I}$ to each template. For each template $\mathcal{T}_i$, we first find an optimal correspondence between the sketch sample points $\mathcal{P}_i$ and the template's exterior points $\mathcal{P}_{\mathcal{T}_i}^{\text{outl}}$. Following Belongie et al. [4], we introduce a set of "dummy" points to model null correspondences. The cost of matching two points is the dissimilarity between their normalized shape context histograms, as measured by the χ^2-test statistic. For dummy points we set a cost threshold θ^{SC} so that any point not matching a legitimate point with cost below θ^{SC} will instead be mapped to a dummy point, indicating a null match.

Given the augmented dissimilarity matrix for the sketch and template points, the optimal correspondence is the permutation of the sketch points minimizing the summed dissimilarity of the matched points. This corresponds to a weighted bipartite

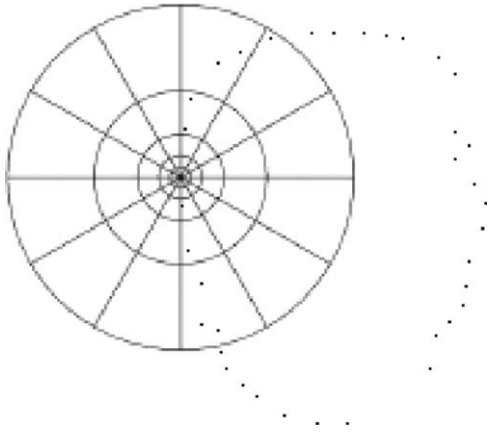

Figure 4 Log-polar histogram template for one point in the skull anterior outline sample of Fig. 3(d).

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

graph matching problem solvable in $O(N^3)$ time using the Hungarian method [14]; following [4], we use the more efficient algorithm of [15].

Once we obtain the optimal correspondence between sketch points $\mathcal{P}_i$ and template outline points $\mathcal{P}^{\text{outl}}_{\mathcal{T}_i}$, we use the full shape context histograms:

$$\boldsymbol{h}(p_j, \mathcal{P}^{\text{outl}}_{\mathcal{T}_i} \cup \mathcal{P}^{\text{int}}_{\mathcal{T}_i})$$

in the computation of the point dissimilarity component of the SC dissimilarity measure. As explained earlier, the outline-only histogram gives us robustness to the level of interior detail when determining point correspondences, and the full histogram gives us sensitivity to the level of interior detail when assessing overall similarity.

Note that the optimal correspondence between sketch points and template points is *rotation dependent*. As an example, consider the skull anterior outline sample points in Fig. 4. If the sample points were rotated a large amount, say 90° (without rotating the histogram template), the resulting shape context histogram would be completely different, preventing correct matching. If, on the other hand, the sample points were rotated a small amount, say 10°, the resulting shape context histogram would barely change at all. For anatomical sketches, this is precisely the desired behavior. Since no sketch will ever be perfect, the technique needs to be tolerant to small deformations; however, since anatomical sketches are always drawn in roughly the same orientation, it would usually be incorrect to match a template point with a sketch point looking locally similar when rotated by a large angle.

In addition to computing the cost of the point correspondences (the sum of the χ^2 differences in SC histograms for each matched point), we also compute the cost of aligning $\mathcal{I}$ with $\mathcal{T}_i$. Following Belongie et al., we estimate a thin-plate spline (TPS) model of the transformation from the sketch points to their corresponding template points. The alignment cost is the "bending energy" of the model [4]. However, whereas Belongie et al. then perform additional iterations of correspondence estimation and alignment, to improve runtime efficiency, we stop after the first iteration. In our case, the SC dissimilarity function, $d^{\text{SC}}(\mathcal{I}, \mathcal{T}_i)$, is simply the sum of the initial point matching costs and the initial TPS model's bending energy cost.

Finally, after computing the dissimilarities $x_i = d^{\text{SC}}(\mathcal{I}, \mathcal{T}_i)$ between sketch $\mathcal{I}$ and each template $\mathcal{T}_i$, UNAS forms the feature vector $\boldsymbol{x}_{\mathcal{I}} = [x_1, \ldots, x_n]^{\mathsf{T}}$, which is then input to the sketch classification SVM $\hat{y} = h(\boldsymbol{x})$. We describe the sketch classification SVM next.

3.1.3. SVM classifier

As previously described, the UNAS sketch classifier is a support vector machine (SVM) $h : \mathbb{R}^{\text{m}} \mapsto \{1, \ldots, N\}$ mapping the feature vector:

$$\boldsymbol{x}_{\mathcal{I}} = [d^{\text{SC}}(\mathcal{I}, \mathcal{T}_1), \ldots, d^{\text{SC}}(\mathcal{I}, \mathcal{T}_{\text{m}})]^{\mathsf{T}}$$

to a category $\hat{y}$. The standard SVM [16] only performs binary classification, so our classifier actually uses several separate SVMs, each separating some classes of sketches from others. We use the ν-SVM [17] and the "one-against-one" multi-class method as implemented by LIBSVM [18] and described in [19]. The idea is, for N classes ($N = 15$ in our case), to train $(N(N-1))/2$ ν-SVMs to separate each pair of classes $\mathcal{I}$ and j. Then, given a new input sketch $\mathcal{I}$ with feature vector $\boldsymbol{x}_I$, we let the "vote" of classifier h_{ij} be

$$h_{ij}(\boldsymbol{x}_{\mathcal{I}}) = \begin{cases} i & \text{if } \sum_k \alpha_{ijk} K_\gamma(\boldsymbol{x}_{\mathcal{I}}, \boldsymbol{x}_{ijk}) - b_{ij} \geq 0, \\ j & \text{otherwise.} \end{cases} \quad (1)$$

Once we have the vote of each classifier, the final classification $h(x_{\mathcal{I}})$ is simply the class that received the most votes. In Eq. (1), k indexes the "support vectors" $\boldsymbol{x}_{ijk}$ for the classifier separating class i from class j. Weights α_{ijk}, b_{ij} and support vectors $\boldsymbol{x}_{ijk}$ are established by the ν-SVM training algorithm [17]. Our classifier uses the radial basis function kernel:

$$K_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{e}^{-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2},$$

in which γ, the kernel width, is a hyperparameter that must be set a priori. The ν-SVM requires one other hyperparameter, ν, which expresses the experimenter's willingness to tolerate incorrectly classified training set items. In the work reported in this paper, we hand-tuned γ and ν to maximize performance on the test set.

Once UNAS obtains the SVM's prediction 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 to find a set of N_s point correspondences $(x_i, y_i) \leftrightarrow (x'_i, y'_i)$ between the sketch and the template. Our current prototype does not distinguish between outline and interior points for purposes of sketch segmentation. The correspondences are then used to fit a thin plate spline (TPS) model [11] 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

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

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. [4] 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 six times. The result is a smooth mapping from every point in $\mathcal{T}^*_{\hat{y}}$ to a point in $\mathcal{I}$.

The canonical templates $\mathcal{T}^*_{\hat{y}}$ 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 (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, solutions to the *traveling salesperson tour problem* (TST) accomplish exactly this task. The goal in the 2D TST problem is to find a shortest-length closed path through a set of points in the plane. Giesen [20] has shown that if the points are samples from a regular closed curve, and the points are sampled densely enough, then the TST correctly orders the points along the curve. He also shows that the edges in the TST under these conditions are also edges in the Delaunay triangulation of the point set. He uses these insights to design an $O(n \log n)$ algorithm that also computes the correct polygonal reconstruction of the curve, again provided that the point set is sampled densely enough.

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 [12] to "connect the dots." The result is a simple polygonal approximation to the boundary of region z_i in $\mathcal{I}$.

The only difficulties with the traveling salesperson approach occur when Giesen's conditions are not met. When insufficiently many corresponding points are found along a contour (the samples are not dense enough), or when an incorrect correspondence is established by shape context (some of the points are not in fact samples from the true closed curve of interest), we obtain noticeably irregular region boundaries and poorly proportioned sketch regions. UNAS incorporates one simple heuristic, *distance-based elimination*, that aims to eliminate incorrect correspondences prior to calculation of the region boundaries.

The distance-based elimination (DBE) heuristic works by eliminating the correspondence between template point (x_i, y_i) and sketch point (x'_i, y'_i) whenever the distance between sketch position $(f_x(x_i, y_i), f_y(x_i, y_i))$ predicted by the current thin-plate spline model and the actual sketch position (x'_i, y'_i) of the corresponding sketch point is above some fixed threshold. Rather than apply some arbitrary distance threshold across all sketches, however, we first z-scale each distance before comparing to the threshold (z-scaling simply subtracts the mean distance over all correspondences then divides by the standard deviation). We find that a DBE threshold of +1.0 standard deviation appreciably improves our segmentation results.

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 an SVM classifier for the standard views of six anatomical structures: brain, heart, lungs, kidneys, stomach, and skull, for a total of 15 different classes of sketches, as shown in Fig. 2. We then evaluated the accuracy of the SVM in classifying sketches. We chose this particular set of anatomical structures to provide a challenging test for the system. It contains some structures with great detail (e.g. heart anterior) and some with little detail (e.g. stomach exterior). It also contains structures that look quite similar (e.g. skull basal and brain basal).

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

We collected sketches of all 15 views from 80 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. We used the 15 drawings shown in Fig. 2, chosen by visual inspection, as templates, i.e. features for the SVM. We then randomly selected 70 sketches for each view from the remaining sketches, for a total of 1050 sketches. For each view we randomly separated the sketches into 70% (49 sketches) for training and 30% (21 sketches) for testing. To determine good ν-SVM hyperparameters, we first fixed ν to its default value of 0.5 and tried various values of γ. For each candidate hyperparameter setting, we trained a set of $(N(N-1))/2$ ν-SVMs and evaluated the resulting classifier on the test set. After obtaining the best value for γ, we fixed its value and searched for a good value of ν. We repeated this procedure until we found a local minimum in the test set error. We evaluated the system with approximately 20 experiments in all. Here we report the best results over all experiments, obtained with $\gamma = 0.0005$ and $\nu = 0.3$.

The SVM classifier had an overall classification accuracy of 75.5% on the test set, far above the baseline random classification accuracy of 6.7%. The accuracy over the various views ranged from a low of 67% for the brain sagittal, heart interior, heart posterior, kidney internal, and kidney external views to a high of 90% for the skull basal view. The detailed results are shown as a confusion matrix in Fig. 5. From the matrix we can see that many of the errors are due to confusing different views of the same structure, e.g. brain sagittal and brain parietal or stomach internal and stomach external.

The system required between 4 s and 7 s to classify each sketch. We found that the execution time was dominated by the estimation of point correspondences in the shape context similarity score calculation (for each sketch, UNAS must estimate point correspondences with every template). One straightforward way to improve the run time would be to reimplement the prototype in an efficient compiled language such as C (currently the SC calculation is implemented in Matlab).

As explained in Section 3.1.2, UNAS separates template sample points into two sets: a set of outline points and a set of interior points. Only the outline points are used for sketch-to-template correspondence estimation. In principle, this technique makes correspondence estimation robust to the

Actual \ Predicted		Brain			Heart			Kidney		Lung		Skull			Stomach	
		Basal	Sagittal	Parietal	Anterior	Interior	Parietal	Interior	Exterior	Interior	Exterior	Anterior	Basal	Parietal	Interior	Exterior
Brain	Basal	16	0	1	1	0	0	0	0	0	0	0	3	0	0	0
	Sagittal	0	14	6	0	0	0	0	0	0	1	0	0	0	0	0
	Parietal	1	5	15	0	0	0	0	0	0	0	0	0	0	0	0
Heart	Anterior	1	0	0	15	2	2	0	0	0	0	0	1	0	0	0
	Interior	1	0	0	2	14	1	0	0	1	0	1	1	0	0	0
	Parietal	1	0	0	1	0	14	0	0	0	0	0	3	0	2	0
Kidney	Interior	0	2	1	1	0	0	14	2	1	0	0	0	0	0	0
	Exterior	0	1	0	0	1	1	2	14	1	0	1	0	0	0	0
Lung	Interior	0	1	0	0	0	0	1	0	18	1	0	0	0	0	0
	Exterior	0	0	0	0	0	1	0	0	3	17	0	0	0	0	0
Skull	Anterior	2	0	0	1	0	0	0	0	0	0	18	0	0	0	0
	Basal	0	2	0	0	0	0	0	0	0	0	0	19	0	0	0
	Parietal	1	0	0	0	0	1	0	1	0	1	0	0	17	0	0
Stomach	Interior	0	0	0	1	0	1	0	0	0	0	0	0	1	17	1
	Exterior	0	0	0	0	0	1	0	0	0	0	0	0	0	4	16

Figure 5 Confusion matrix for the SVM classifier run on 21 sketches for each of 15 views.

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

Actual \ Predicted		Brain			Heart			Kidney		Lung		Skull			Stomach	
		Basal	Sagittal	Parietal	Anterior	Interior	Parietal	Interior	Exterior	Interior	Exterior	Anterior	Basal	Parietal	Interior	Exterior
Brain	Basal	205	1	0	0	0	0	0	0	0	0	0	4	0	0	0
	Sagittal	1	195	14	0	0	0	0	0	0	0	0	0	0	0	0
	Parietal	0	15	195	0	0	0	0	0	0	0	0	0	0	0	0
Heart	Anterior	0	0	1	183	5	20	0	0	0	0	0	0	0	0	1
	Interior	0	0	0	9	197	2	0	0	2	0	0	0	0	0	0
	Parietal	2	1	1	29	8	169	0	0	0	0	0	0	0	0	0
Kidney	Interior	0	0	0	0	0	0	203	6	1	0	0	0	0	0	0
	Exterior	0	0	0	0	0	0	4	204	0	1	0	0	0	0	1
Lung	Interior	0	0	0	0	0	0	1	1	191	17	0	0	0	0	0
	Exterior	0	0	0	0	0	0	1	0	13	196	0	0	0	0	0
Skull	Anterior	0	0	0	0	0	0	0	0	0	0	208	2	0	0	0
	Basal	5	0	0	0	0	0	0	0	0	0	0	205	0	0	0
	Parietal	0	1	2	0	0	0	0	0	0	0	0	0	207	0	0
Stomach	Interior	0	0	0	0	0	0	2	0	2	0	0	0	0	196	10
	Exterior	0	0	0	0	0	0	0	0	1	0	0	0	0	8	201

Figure 6 Confusion matrix for the classification judgment of 10 sixth-year medical students. The sample size for each view is $10 \times 21 = 210$.

level of detail in the sketch. Better point correspondence should in turn improve classification performance. To assess the effect of the technique empirically, we reran the evaluation without separating the template sample points into separate outline and interior point sets. Under this condition, UNAS obtained a classification accuracy of 72.0%, compared to 75.5% with interior point removal. The modest improvement due to interior point removal suggests the importance of accurate point correspondence estimation for classification accuracy. It also indicates that further improvement might be obtained if we eliminate the sketch's interior points automatically prior to correspondence estimation, as previously discussed. Interior point removal would have the additional benefit of improving UNAS's run time performance.

To provide a reference against which to compare the accuracy of the SVM, we asked 10 sixth-year medical students to classify the sketches in the test set. In order to make the comparison with the SVM fair, we used a forced classification, permitting them to choose only among the 15 classes. The mean classification accuracy over the 10 students and 15 views was 93.8%. The accuracy over the various views ranged from a low of 80.5% for heart anterior view to a high of 99.0% for the skull anterior view. The high value for the skull anterior view is not surprising since people are particularly good at recognizing faces. The confusion matrix for the group of 10 students is shown in Fig. 6. We can see that while the error rate for the students is lower than for the SVM, the types of errors are similar. For example, the students also often confuse the brain sagittal and brain parietal views and the stomach internal and stomach external views.

Finally, 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, the parietal view of the brain, the anterior view of the heart, and the parietal view of the skull. We chose three qualitatively different sketches for each organ. Each sketch used was correctly recognized by UNAS. The segmentation results are shown in Fig. 7. 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.

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

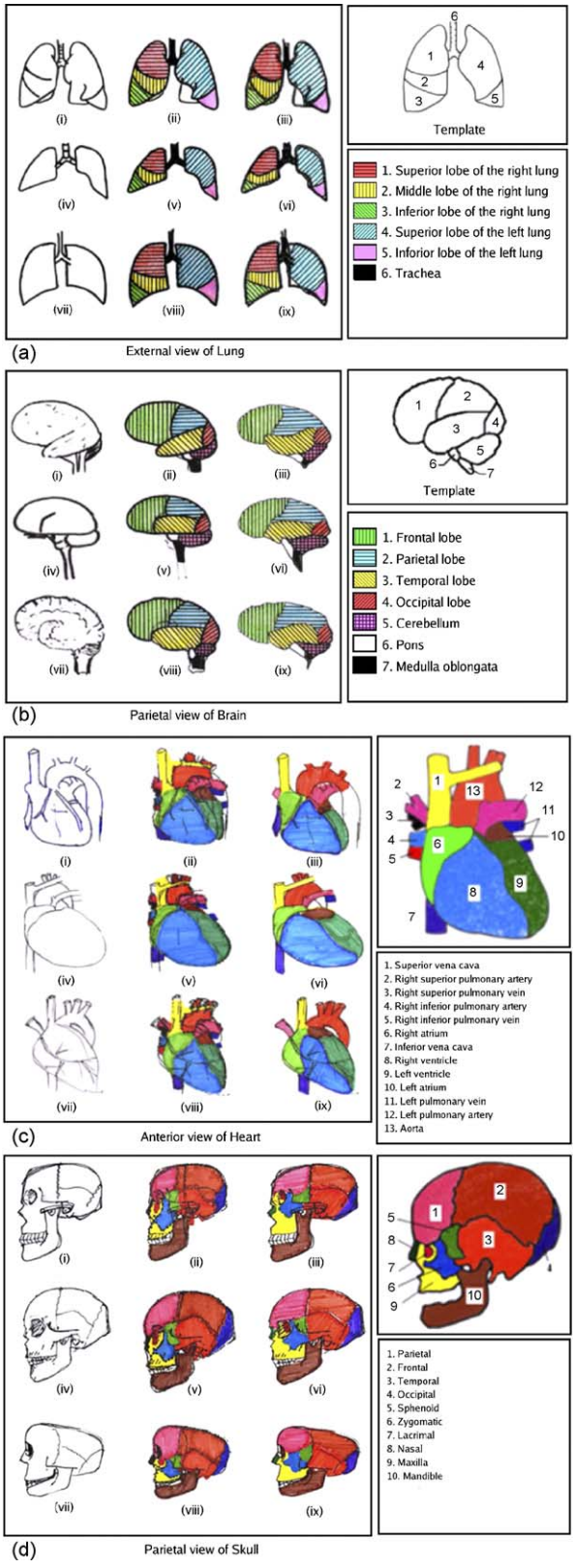

Figure 7 Medical student sketches (first column) with corresponding segmentations produced by a physician (second column) and by UNAS (third column). The templates used by the segmentation algorithm are shown in the upper right corners.

Although the UNAS sketch segmentation system performs very well when a given sketch is similar in proportion and detail to the canonical template used as a reference, differences between UNAS and the human expert arise when sketch details are incorrect or missing. In these cases, UNAS is sometimes unable to establish correct point correspondences, causing distorted region boundaries compared to the expert-drawn boundaries. We find that UNAS makes two main kinds of mistakes. The first type occurs when the sketch contains a line, intended as a region boundary, that is in an incorrect position relative to the rest of the sketch. Under these circumstances, the shape context histograms may be so dissimilar for points on the sketch line and putatively corresponding template line that they are not matched. Without being guided by point correspondences for a given boundary, the thin plate spline transformation can only map the boundary points from the template to the "correct" position in the sketch, effectively correcting the user's incorrectly drawn boundary. The physician's boundaries, in contrast, normally follow the lines in the sketch as long as they are not too far off. This can be seen by comparing the superior lobe of the left lung in segmentations (viii) and (ix), and the cerebellum in segmentations (viii) and (ix). The second type of error occurs when one region boundary in the canonical template needs to be distorted in order to align with the corresponding boundary in the sketch. Since the thin-plate spline transformation is global, distortion of the first boundary will exert a distorting force on other neighboring boundaries unless there are correct point correspondences to resist the distortion. This is particularly troublesome for very small parts, on whose boundaries UNAS samples few points, leading to fewer correct correspondences and potentially an insufficiently constrained thin-plate spline mapping. One example in Fig. 7 is the pons in segmentation (vi).

5. Conclusions and future research

The performance of the UNAS prototype provides strong evidence that development of robust sketch understanding systems for medical domains is an attainable goal which could have significant implications for effectiveness of medical tutoring systems and usability of electronic patient records. While the results from our prototype system are encouraging, much work remains to be done in order to realize the functionality described in our motivating example. First the accuracy of the recognition and segmentation algorithms could be improved. More accurate recognition could likely

Please cite this article as: Peter Haddawy et al., Anatomical sketch understanding: Recognizing explicit and implicit structure, Artificial Intelligence in Medicine (2006), doi:10.1016/j.artmed.2006.07.010

be obtained by explicitly detecting the outline of the sketch and then using that to match the outline of the template rather than using dummy points. This should give a more accurate match between the two outlines. The results of the segmentation algorithm could be improved by using local transformations to fit the template to the sketch. We have obtained good preliminary results in this direction with Rangarajan and colleagues' robust point matching algorithm [21,22].

In our prototype we have assumed that a sketch includes only one anatomical structure, but sketches often contain multiple structures as well as incompletely drawn structures. A fully general sketch recognition system should be able to handle this. Extending our approach to cover this might require adding spatial reasoning.

In addition to understanding the sketch, UNAS should be able to recognize and interpret annotations commonly used in medicine, such as arrows, circles, crosses, darkened regions, and clusters of dots. For the recognition we are exploring the use of hidden Markov models, which tend to work well for such relatively simple symbols. The interpretation of such symbols is more difficult. A cluster of dots drawn on a sketch of an arm might indicate a rash but the same symbol drawn on a sketch of the sinuses might indicate a diffuse area of infection. Interpreting such annotations requires background knowledge. We are exploring the use of the UMLS medical ontology to provide such knowledge by linking anatomical parts with commonly associated pathology and physiology. The final step will be to incorporate UNAS as part of our COMET intelligent tutoring system and evaluate its impact on student learning outcomes.

Acknowledgments

Funding for this work was provided by the Royal Thai Government. We thank Dr. Siriwan Suebnukarn for generously providing her time and expert medical advice.

References

[1] Suebnukarn S, Haddawy P. A collaborative intelligent tutoring system for medical problem-based learning. In: Johnson L, Andre E, editors. Proceedings of international conference on intelligent user interfaces. New York: ACM Press; 2003. p. 14–21.
[2] Suebnukarn S, Haddawy P. A Bayesian approach to generating tutorial hints in a collaborative medical problem-based learning system. Artif Intell Med 2006;38(1):5–24.
[3] Suebnukarn S, Haddawy P. Clinical-reasoning skill acquisition through intelligent group tutoring. In: Kaelbling LP, Saffiotti A, editors. Proceedings of international joint conference on artificial intelligence. Denver, CO: Professional Book Center; 2005.
[4] Belongie S, Malik J, Puzicha J. Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 2002;24(4):509–22.
[5] Hearst MA, Gross MD, Landay JA, Stahovich TE. Sketching intelligent systems. IEEE Intell Syst 1998;13(3):10–9.
[6] Gross MD. The proverbial back of an envelope. IEEE Intell Syst 1998;10–3.
[7] Landay JA, Myers BA. Sketching interfaces: toward more human interface design. IEEE Comput 2001;56–64.
[8] Alvarado CJ, Davis R. Resolving ambiguities to create a natural sketch-based interface. In: Nebel B, editor. Proceedings of international joint conference on artificial intelligence. San Francisco: Morgan Kaufmann; 2001.
[9] Forbus KD, Usher J. Sketching for knowledge capture: a progress report. In: Gil Y, editor. Proceedings of international conference on intelligent user interfaces. New York: ACM Press; 2002. p. 71–7.
[10] Abate AF, Nappi M, Tortora G, Tucci M. Assisted browsing in a diagnostic image database. In: Catarci T, Costabile MF, Levialdi S, Santucci G, editors. Proceedings of workshop on advanced visual interfaces. New York: ACM Press; 1996. p. 223–32.
[11] Bookstein F. Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans Pattern Anal Mach Intell 1989;11(6):567–85.
[12] Gulley N. Traveling salesman problem demonstration (MATLAB demo source code); 1993. Software available at: http://www.mathworks.com (accessed on 27 May 2006).
[13] Otsu N. A threshold selection method from gray-level histogram. IEEE Trans Syst Man Cybernet 1979;9:62–6.
[14] Papadimitriou C, Stieglitz K. Combinatorial optimization: algorithms and complexity Englewood Cliffs, NJ: Prentice Hall; 1982.
[15] Jonker R, Volgenant A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing 1987;38:325–40.
[16] Vapnik VN. The nature of statistical learning theory New York: Springer; 1995.
[17] Scholkopf B, Smola A, Williamson R, Bartlett PL. New support vector algorithms. Neural Comput 2000;12:1207–45.
[18] Chang C-C, Lin C-J. LIBSVM: a library for support vector machines, 2001. Software available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed on 2 September 2006).
[19] Hsu C-W, Lin C-J. A comparison of methods for multi-class support vector machines. IEEE Trans Neural Networks 2002;13:415–25.
[20] Giesen J. Curve reconstruction in arbitrary dimension and the traveling salesman problem. In: Bertrand G, Couprie M, Perroton L, editors. Proceedings of discrete geometry and computational imagery conference. Berlin: Springer; 1999. p. 164–76.
[21] Gold S, Rangarajan A, Lu C-P, Pappu S, Mjolsness E. New algorithms for 2D and 3D point matching: pose estimation and correspondence. Pattern Recogn 1998;31(8):1019–1031.
[22] Chui H, Rangarajan A. A new algorithm for non-rigid point matching. In: Kriegman DJ, Forsyth DA, editors. Proceedings of IEEE conference on computer vision and pattern recognition, vol. 2. Los Alamitos, CA: IEEE Computer Society; 2000. p. 44–51.