Anoto Medical Image Annotator: Interactive Prototype and Progress Report

1. Member Names and Roles

Group Name: The Annototators
Anirudh Vemprala: User interface design and programming
Edward Karuna: User interface design, programming, and testing.
Gene Zhang: User interface design, programming, and testing.

1. Problem and Solution Overview

Radiologists and medical imaging researchers work in a field that demands constant, careful evaluation of complex images. The ability to accurately evaluate a medical image and detect important features, whether they are broken bones or metastasized tumors, is tantamount to success in radiology. Therefore, it is important for radiologists and researchers to have tools that complement their abilities by making it easy to record observations and sharing them with others. Currently, radiologists have limited on-screen tools to select regions of interest and share their observations with colleagues. Most people review medical images by opening the files on their respective computers, making personal notes, and then presenting their observations in a group setting. The solution to the current, inefficient system is an integrated approach that allows one to attach comments to a medical image and permit others to add their own observations. The spirit of collaboration could also be extended to include a student-teacher relationship. That is, students could indicate what they believe to be important features within an image, and then the teacher could review the students’ work, indicate missed features, and provide tips for future analyses. The solution should also include the ability to create the observations directly on a printout, allowing the user to use a pen to circle features and jot notes and avoid the relatively clumsy use of a mouse for the same actions. The Anoto digital pen and its range of abilities are well-suited for such a solution. We have named our solution the “Anoto Medical Imaging Annotators (AMIA)”.

2. Tasks

The following tasks outline the key functionality provided by our system.

Task 1 (Easy): Login and load previous annotations for viewing.
In order to facilitate the sharing features of the program, users will need to be uniquely identified. The user login will also be crucial for the teacher-student relationship, since students will not be able to view their peers’ reviews of a given image if the program is being used to evaluate their performance. The teacher, on the other hand, would be given access to everyone’s files in order to deliver his/her comments.
When the program loads up, a sign-in dialogue box will appear and ask the user for his/her screen name and password. The task then demands that the user load a previous set of
annotations, an action which will be necessary in order to add comments to other users’ notes for a given set of medical images. The task should be a straightforward process.

**Task 2 (Medium): Add text comments to another person’s annotations**
To collaborate on the review of patient's medical images, multiple users must be able to comment on the same images. The program includes a text box for providing additional comments on top of those already recorded using the anoto pen. The task involves selecting an image with pre-existing comments, selecting the text box, adding a few lines of text, and committing the changes to the file using an “add comments” button.

**Task 3 (Hard): Print and annotate a set of image.**
The core functionality of AMIA lies in its use of the anoto digital pen. Therefore, it was crucial to include a task that required the user to print out a set of medical images, add comments, and register those comments with the program. The task is accomplished by reviewing a medical image set, clicking the “Print” button, selecting the images to be annotated, retrieving the printed images, using the pen to indicate comments and various shapes on the images, and then verifying that the strokes were loaded into the computer and associated with the correct images.

3. Revised Interface Design

**Results of Contextual Inquiry**
We learned from the contextual inquiry assignment that our target users (radiologists and medical imaging researchers) would like a quick, intuitive way to choose regions of interest (ROI’s). Our target users work with many medical images on a regular basis, so any device or system that can improve efficiency and reduce the time spent on each image would be welcomed. Additionally, a system for the simultaneous review of resident sample analyses of images would enhance an instructing radiologist’s ability to assess their findings and return helpful comments. One interviewee emphasized her desire for a quicker, more direct way to annotate medical images in general. We also learned that radiologists typically assess images in a dedicating reading room that includes a handful of computer stations. They also share their findings with their patients in their offices, and may verbally share findings in group meetings. Researchers process medical images at the point of acquisition, at lab workstations, at their desktop computers, and even on their laptops. Sharing of written observations was therefore lacking. Based on the preceding information, the interface we had in mind was one that resembled a multi-document image viewer. We thought it would be wise to include a preview pane, would allow quick selection of image sets from within a given directory. The main document space would present its user with the medical images under consideration, along with representations of layers that store the annotations made to the images. A slidebar at the bottom of the screen would allow the user to rapidly scan through the slices within the imaging set. For a given image, a set of layer thumbnails would be displayed near the bottom of the screen. Each layer would represent comments from a different user. By selecting one of the thumbnails, one could place that layer over the original image in the main window. A "Comment" button would then create a dialog box where the user can add his/her comments about the currently viewed layer. These comments would then be attached to that layer's metadata, and would be viewable by the
original creator of that layer. Additionally, a "Print" button would be included. After selecting the print button, the user would be prompted to print either an ROI printout or a commenting printout on Anoto paper. A commenting printout would only include the original image and a "Done" checkbox. An ROI printout would include the original image and "Done" box, along with checkboxes for running certain types of analysis, such as pixel histograms, once the Anoto strokes are uploaded to the computer. Figure 1 indicates our initial concepts.

Figure 1: Left to right: Rough sketch of computer interface, paper UI for general annotation, paper UI for ROI selection.

Lo-Fi Prototype Design
Following the contextual inquiry, we created a pen, paper, and tape low-fidelity implementation of our interface. Figure 2 shows an example screen, while Figure 3 shows the lo-fi paper interface for general comments.

Figure 2: Lo-fi computer interface.
As seen above, we kept our ideas for a comment text box, a main window showing the currently selected image, a thumbnail slider for images, and button shortcuts for actions such as printing, saving, exporting, and opening. The left side of the screen also includes collapsible lists (collapsed in the presented view) that allow the user to toggle various Anoto-created comment layers and ROI sets associated with the currently viewed image. We had the users accomplish three tasks: jot down comments on a general comments printout, run ROI analysis using an ROI printout, and add comments to another user’s annotations using the comments text box. Figures 3-5 show the storyboards for the tasks.

**Storyboard for Resident commentary using AMIA**

Figure 4: General annotation task.
Lo-Fi Prototype Reactions
After testing the lo-fi prototype with three target users, we gained valuable insight into the advantages and disadvantages of AMIA. First, we learned that the ROI selection functionality should be removed. Upon examination of our users' interaction with this feature, we learnt that not only were they having some difficulty achieving the required tasks, but they also found the use of paper-based medium for selection regions of interest rather unhelpful. Given that there are existing tools that can do ROI selection in an automated fashion with more sophistication than AMIA, it was decided to focus in on the collaborative/instructive aspect of the program as there are no tools that provide such functionality in the market today.
Additionally, users found the "Add" button (meant to represent Add Comment for the on-screen text box) rather confusing. We decided to rename it to "Save Comment".

Some users expressed concerns with the way file I/O was conducted through the UI. The prototype had a persistent "save file" pane which was confusing to users. Some suggested moving this into a more traditional File->Save or Save As action. Users also recommended not launching the application with an "Open File" dialog, but rather allow them to manually open files. To that end, features like thumbnails in the file dialog boxes would be helpful as our application deals with images. However, we considered these to be a bit too advanced for our interactive prototype and have left the file open process as mostly a “wizard of oz” feature for the time being.

Medical image orientation was considered important to the interviewees. All of our examples include MR coronal cross-sections of the brain, when in reality the user would be able to view sagittal, coronal, and transverse sections of the same patient. We decided not to address the issue in the interactive prototype, as coronal images are sufficient to test the program’s functionality.

Finally, the lo-fi prototype of AMIA required the user to print out slices of an image one at a time. Based on the prototype test, we learned that it would be easier for the user to select multiple slices and print them at once on Anoto paper.

4. Prototype Overview

Overview of implemented UI

When the program is executed, the user is first greeted with a login screen. After successfully entering a username and password, the main interface opens (see Figure 7). Currently, only dummy images are displayed. The user must click the “Open” button and enter the name of an image set, which is then used by the program to find and load the images into the system. Clicking the “Print” button allows the user to print out a set of medical images on individual pages. Each printout (Figure 8) includes anoto dot patterns that allow the user to draw shapes on the image, write down comments, and indicate when he/she is done annotating the printout. Currently, the “done” checkbox is not implemented because we have chosen to use the streaming functionality of the anoto pen. The shapes and comments are shown on the screen over and below the medical image, respectively. The user can also select the text box and add text comments to be associated with the image. Please refer to Section 3 for the specific tasks demonstrated during the presentation.
Figure 7: Screenshot of user interface. The main window displays the current image, while the thumbnail browser across the top can be used to switch between images within a set. The buttons along the right will allow the user to open, save, print, and export image. The text box on the bottom provides a way for the user to enter text comments to be associated with the current image.

Figure 8: Example paper user interface. The anoto dot pattern is printed over the medical image to facilitate annotation. The comment box is used to indicate additional notes. The check box is used to tell the system that the current annotation session is over.
Omitted Features and Wizard of Oz Replacements
As stated in the lo-fi prototype reactions section, we found it necessary to omit several features for the interactive prototype. We did not implement transverse, coronal, and sagittal views of the imaging sets. We did not have access to all three views of an image set and decided that coronal views would be enough to demonstrate the general functionality of the program, so we left out the other two. The ability to navigate between folders within a file system while saving or loading was not included in the program due to time constraints. Thus, we decided that a Wizard of Oz (WoZ) approach would be sufficient for file saving and loading. WoZ techniques were also used for image printing to more efficiently direct our focus on coding on how the program deals with anoto strokes. Therefore, during the interactive prototype demonstration, the user is not given full control over which images can be printed out and annotated. Rather, a set of images were pre-printed to streamline the demonstration. This was considered to be a crucial decision, as the printing of anoto paper UI's can take an excessive amount of time.