Published in final edited form as:
Proc SPIE Int Soc Opt Eng. 2011 March 3; 7966: . doi:10.1117/12.878412.

Characterizing and Optimizing Rater Performance for Internet-based Collaborative Labeling

Joshua A. Stein[*,a], **Andrew J. Asman**[a], and **Bennett A. Landman**[a,b]

[a]Electrical Engineering, Vanderbilt University, Nashville, TN, USA 37235

[b]Biomedical Engineering, Johns Hopkins University, Baltimore, MD, USA 21218

Abstract

Labeling structures on medical images is crucial in determining clinically relevant correlations with morphometric and volumetric features. For the exploration of new structures and new imaging modalities, validated automated methods do not yet exist, and so researchers must rely on manually drawn landmarks. Voxel-by-voxel labeling can be extremely resource intensive, so large-scale studies are problematic. Recently, statistical approaches and software have been proposed to enable Internet-based collaborative labeling of medical images. While numerous labeling software tools have been created, the use of these packages as high-throughput labeling systems has yet to become entirely viable given training requirements. Herein, we explore two modifications to a typical mouse-based labeling system: (1) a platform independent overlay for recognition of mouse gestures and (2) an inexpensive touch-screen tracking device for non-mouse input. Through this study we characterize rater reliability in point, line, curve, and region placement. For the mouse input, we find a placement accuracy of 2.48±5.29 pixels (point), 0.630±1.81 pixels (curve), 1.234±6.99 pixels (line), and 0.058±0.027 (1 – Jaccard Index for region). The gesture software increased labeling speed by 27% overall and accuracy by approximately 30-50% on point and line tracing tasks, but the touch screen module lead to slower and more error prone labeling on all tasks, likely due to relatively poor sensitivity. In summary, the mouse gesture integration layer runs as a seamless operating system overlay and could potentially benefit any labeling software; yet, the inexpensive touch screen system requires improved usability optimization and calibration before it can provide an efficient labeling system.

Keywords

Observer Performance Evaluation; Technology Assessment; Parcellation; Labeling; Human Machine Interaction

1. INTRODUCTION

The labeling or parcellation of structures of interest on magnetic resonance imaging (MRI) is crucial in determining correlation with clinically important morphometric and volumetric features. For neuroimaging labeling tasks with well developed method, some automated

[*]joshua.a..stein@vanderbilt.edu; http://masi.vuse.vanderbilt.edu; Medical-image Analysis and Statistical Interpretation Laboratory, Department of Electrical Engineering, Vanderbilt University, Nashville, TN, USA 37235.

approaches have been shown to significantly outperform even expert human rater [1]. In the mapping of new structures and new imaging modalities such as with whole-brain diffusion tensor imaging, our knowledge is evolving and automated approaches have not yet been developed and/or validated (e.g. [2]). For these cases, labeling through expert tracing remains the gold standard for mapping intricate and perhaps subtle anatomical divisions, such as with the hippocampus [3] or cerebellum [4], and even for finely detailed whole brain parcellations [5]. The process of voxel-by-voxel labeling from an individual rater can be extremely time consuming and resource intensive. Statistical methods have increased the field of potential raters for highly accurate labeling from expert neuroanatomists to include minimally trained human raters [6, 7].

Even for collaborative labeling, efficient, accurate, and precise labels are essential for successfully achieving large scale efforts. While numerous labeling software tools have been created, the use of these packages as high-throughput labeling systems has yet to become entirely viable. With the increased audience introduced by the statistical methodologies, the ability to handle large scale label sets and rater numbers becomes paramount in determining the viability of a software labeling tool.

When we consider a person as an “image processor”, the human machine interface becomes an apparent limiting factor of efficiency and performance of a collaborative labeling effort. Humans have a seemingly vast computational ability in terms of vision and visual analysis, and this is worth exploring as long as the divide between human and machine computational power remains large. Humans can discern shape, color, and extract relevant meaning from contextual information embedded in an image on the order of milliseconds. However, it may take over a minute to label even a few simple shapes for later parsing by a computer. In addition to the time bottleneck, the accuracy and precision with which a human can send data to the computer may be limited by the interface they use. For example, the computer mouse is a simple ubiquitous piece of hardware that is designed for pointing, but is also often used for path-tracking, and is used as such in labeling applications. However, a more specialized input device designed explicitly for path-tracking may help humans to increase labeling efficiency and accuracy in vision related tasks. Because human computational power is leveraged for image analysis in a variety of clinical and scientific endeavors, it is relevant to explore interface optimizations between the human and the machine, specific to the task of labeling images.

Herein, we explore two modifications to a standard mouse-based labeling system. First, we implement a platform independent overlay for recognition of mouse gestures to accelerate tool switching. Second, we implement and evaluate an inexpensive (approximately $10) touch-screen tracking device for non-mouse input. This device could enable close interaction with common computers and could be relatively ubiquitous to leverage the power of many minimally trained raters over the web. We approach this challenge with the mindset that these devices could help direct future research in tracking-specific devices.

2. METHODS

Imaging labels were collected with the WebMill system (https://brassie.ece.jhu.edu/Home; NITRC project ID: http://www.nitrc.org/projects/webmill; shown in Figure 1). This open source, custom designed web service presents images to raters, allows raters to label the images and submit their results and labeling information to a remote server using a web browser. After informed written consent, 19 healthy individuals were asked to spend approximately ten minutes per task on three different labeling techniques: (1) A standard mouse: User was responsible for setting the correct pen tool size, color and shape for each task. (2) A standard mouse using GesTr: "hot spots" located at the corners of the screen automatically set pen tool size, color and shape. (3) Infrared pen (Penteractive, Soddy Daisy, TN) and Nintendo Wii Remote (Redmond, Washington) with GesTr: Used infrared pen instead of mouse in conjunction with screen corner "hot spots". Tasks were performed in random order for each participant.

For each labeling technique, the user was responsible for drawing various shapes using a variety of colors and tools. This is very much in line with neuroimaging WebMill tasks, which have users frequently switch tools and colors to label different biological structures in an image in order for the label data to be easily parsed by a computer. In this experiment, the user was responsible for drawing straight lines following a path (a triangle), drawing curved lines following a path (a spiral), marking points (the 5 points on a star), and filling in a shape (an ellipse). For each drawing task the user was told to use a particular color, brush size, and brush shape.

For the standard mouse technique (1), users were given the standard WebMill instructions. For GesTr with the mouse (2) and Infrared Pen / Wii Remote (3), users were told briefly how to change tools by moving the mouse (or pen) into hot spots in the corners of the screen. Additionally, with the infrared pen, users were told how to control the mouse position on the screen and the action of clicking by activating the light on the pen while the pen was in close proximity to the screen and facing the Wii Remote Camera. It is important to note these changes have an effect both on the workflow of the labeler as well as the complexity of the task. There are four gestures for this experiment corresponding to the particular labeling tasks required in this experiment. However, the GUI for WebMill is not dynamic and presents a variety of tools to the user because WebMill is designed to handle a variety of labeling requirements. This is not a limitation of WebMill, but rather a limitation of graphical user interfaces in general. There is a distinct difference in the methodological complexity of a task when using a GUI (or keyboard shortcuts) versus using mouse, because the number of user choices is vastly reduced in the case of the gestures. Additionally, the amount of effort required to code specialized GUIs for subtasks in the domain of rater labeling is very large compared to the effort required to specify gesture actions in XML using GesTr.

3. DATA

Although WebMill was designed to label neuroimages, in this study all anatomical knowledge was abstracted away and we focused solely on evaluating the input devices. The user was shown a gray-scale image on the left and asked to color it as shown in the right. Imaging data was artificially created to test the minimally trained humans' ability to perform various common labeling tasks. Each labeling task (402×402 pixels) had four components each located in one quadrant of the image: (1) Label the five corner points of a star: Tested the raters' ability to label individual points. (2) Fill in an elliptical region: Tested the raters' ability to quickly and efficiently fill a region of interest. (3) Trace the contour of a triangle: Tested the raters' ability to trace straight lines. (4) Trace the contour of a spiral: Tested the raters' ability to trace curves. For each labeling task, the position of a task in a quadrant and the orientation (rotation of 0°, 90°, 180°, or 270°) was randomized, but the brush size, color, and object size remained unchanged.

4. RESULTS

4.1 The Mouse

The mouse was taken to be the baseline of performance because the mouse is used almost universally, even for specialized path-tracking labeling tasks.

4.2 GesTr

The GesTr tracking software improved performance and reduced error compared to the mouse interface without GesTr. Overall, error was reduced by approximately 33% compared to the mouse, while also reducing the time spent on each image by approximately 27%.

4.3 Infrared Pen with the Wii Remote

The Wii remote and pen interface reduced performance and increased error significantly considering both tasks that involved path-tracking and those which involved point-marking. This method resulted in a 37% increase in time spent per image, and a 70% increase in error. Ultimately, the Wii Remote in conjunction with an infrared pen was unable to outperform the mouse, alone or with GesTr, given GesTr's implementation of Wii Remote support along with the infrared pen used.

4.4 Per Person Normalized Results

When the results are viewed per person normalized, every labeler is weighted equally, instead of every label, as above. These results are (expectedly) slightly different.

The mean time reduction normalized per person (between the mouse and GesTr with the mouse) was 18% [p = .39].

The mean error reduction normalized per person (between the mouse and GesTr with the mouse) for point marking was 53% [p = .31], for straight lines: 25% [p = .36], for curved lines: 43% [p = .40], and for the ellipse: −0.5% [p = .39].

These results are inconclusive. The reason they are less conclusive than the non-normalized results is that there is simply less person comparison data than label comparison data, since each person was responsible for many labels.

5. DISCUSSION

The GesTr software itself is relatively straightforward, although it contains some powerful features. At its core, the goal of GesTr is to interface with the operating system (not a particular application) and map gestures to customizable actions, which include combinations of mouse clicks and keyboard keystrokes. It is written in pure Java and leverages a variety of open source, free, cross platform tools with the goals of remaining free and open source itself, while also maintaining freedom from platform. For tasks which require machine specific knowledge, such as polling for screen size, polling the mouse, and applying actions, GesTr interacts with the Java Virtual Machine, which is written specifically for various operating systems and provides hooks into operating system functionality for all common platforms. GesTr is designed to run as a Java Web Start application, and it leverages simple XML scripting to customize the actions that it supports. The implication is that mouse gestures may be added to projects to increase user efficiency with marginal coding effort.

The design choice to run as a Java Web Start application and to interact the operating system directly instead of being used as a tool to integrate into applications on a case-by-case basis was made to reduce the effort necessary to incorporate GesTr's features into well-developed applications. All that is required to incorporate GesTr is to run the application from the web, in a browser, and provide an XML file describing the map of gestures to actions that are required for the particular task at hand. Some of GesTr's long term goals, such as a better implementation and increased support for the Wii Remote with an Infrared Pen, exist under the mindset of finding ways to increase performance collaborative minimally trained labelers at minimal cost. The aforementioned design choices reflect that sentiment. It should be easy to launch GesTr, there should be minimal software requirements (the Java Virtual Machine is a very established software available on most work and personal computers), and it should be as trivial as possible to integrate into a project.

GesTr also incorporates experimental support for use with a Wii Remote and Infrared pen, to generate a low cost solution for adding touch screen functionality to large displays. As of August 2010, a 15“ resistive touch screen display costs approximately $400. Better capacitive touch screen displays cost approximately $600 at 15”. Larger displays with higher resolutions (1600×1200, 20“+) cost on the order of $1000. This cost is much higher than a modern display without touch functionality. The approximate cost of a 24” display at 1920×1200 resolution is $200. The true cost of adding a Wii Remote and Infrared pen to a computer display may be marginal in many cases depending on the availability of a Wii Remote. The Remote generally costs approximately $40 and an Infrared pen cost about $10. Because of the proliferation of the Nintendo Wii, and correspondingly Wii Remotes, the average cost of such an addition is likely much less than $50. If, in the future, high quality, natural, usable touch screen functionality can be added to modern displays using such hardware, it may open the avenue for a dramatic increase in quality of labels and rater efficiency for large scale collaborative labeling projects.

GesTr's current Wii Remote support leverages two separate open source libraries called BlueCove and WiiRemoteJ to handle the non-standard Bluetooth based connection and data polling, respectively, between the Wii Remote and the computer. The Wii Remote captures infrared light at 740 nm and is able to report that light as an (x,y) coordinate relative to the camera's vision. It is important to note that the camera must be placed at an angle to the screen (of approximately 45 degrees). One must attempt to satisfy several conditions with the camera placement to make the system as usable as possible. First, the camera must have an unobstructed view of the light coming from the pen whenever the pen is on the surface of the display. This means that the camera has to be far away enough from the display so that it can view the entire display. Additionally, it means that for right handed users, the Wii Remote's camera should face to the right from the left side of the display, and vice versa for left handed users. Second, the camera must be placed as normal as possible to the screen, so that drastic changes in resolution do not occur from one side of the display to the other. Such behavior may cause a noticeable drop in tracking accuracy if resolution drops sufficiently. This requirement somewhat conflicts with the first requirement in cases where the user may end up blocking the view of the camera, or the angle range of light emission from the pen is insufficient to reach the camera when the pen is held naturally. This, of course, depends on the particular hardware used.

If the user marks the corners of the screen with the infrared light from the pen for the Wii Remote Camera, then, from the perspective of the Wii Remote, four corners of a quadrangle have been defined. GesTr records this quadrangle, polls the operating system for the screen resolution, and calculates the aspect ratio of the screen. At this point, GesTr applies linear algebra to map infrared coordinates within the defined quadrangle (as seen by the perspective view of the rectangular display by the Wii Remote Camera) to pixel coordinates on the display. It uses this information in conjunction with a low pass filter to refer control of the mouse to the Infrared pen. This also relies on the Java Virtual Machine to provide hooks into the operating system in a platform independent manner.

The results of the experiment are very revealing. First, it is clear that GesTr improves the speed of labelers using WebMill. GesTr may also have a positive effect on the quality of label results. Further experimentation would be required to verify this for most cases, although it is likely true for at least point marking based on the trials. Conceptually, the first result makes immediate sense, and is the most important. GesTr was designed with the intention and goal of increasing the speed of the labeling process. However, the underlying reasons and the implications behind the second result are not immediately evident. It may be an indication that the labelers are able to stay slightly more focused on the actual task if "task switching" itself requires a sufficient degree of focus. This explanation is anecdotal only but it does bring to light that further exploration may be valuable in designing a system with the psychology of the user in mind. It may be beneficial to make changes to the interface with the mindset that they may have an effect not only on the user's ability, but also on their behavior.

The other result is that introducing the Wii Remote and Infrared pen as a human machine interface to those unfamiliar with it resulted in poor performance compared to the mouse. Clearly, this tool is not ready for integration into collaborative labeling given its current implementation. However, there are many vectors for improvement on the current implementation that may make the solution viable, which are detailed in the conclusions below.

Another point concerning the results is whether or not to consider the resultant labels as independent events or to tie them to the labelers to see how they improve. Either metric may prove to be more applicable depending on the scenario. In the case where you can indentify and rate the labelers themselves, it seems that it is most pertinent to consider the improvements normalized per person. In the case where you do not weight the data based on the labeler who created it, it makes sense to look at the overall performance change of the population of labels. Another important point to consider is that there are scenarios where changing only speed may have an effect on the average quality of labels, especially if good raters have a tendency to improve their speed to a different degree than worse raters. That would happen if analyzing the labels independently, and low error labels occur more frequently because good labelers were able to leverage GesTr better as well, and therefore generated more labels in the GesTr trials than weaker labelers. The per person normalized results suggest that this is a possibility – another anecdotal explanation of why quality may have improved with GesTr, backed with some data (p values are still insufficient to make any bold claims).

6. CONCLUSIONS

As we explore ever more subtle anatomical correlations in health and disease, we must look towards *efficiently* acquiring increasing amounts of data and make best use of this information. While WebMill, STAPLER, and other statistical approaches have begun to make crowd-sourcing possible for medical image labeling, we must continue to look towards optimizing use of the efforts of our volunteers. From these studies we see that labeling precision is highly variable both with the task at hand as well as with the input device used. Perhaps not surprisingly, use of the GesTr software improved labeling speed by nearly 30%, but we also observed a reduction of error on the point, curve, and line tasks.

Additionally, the negative impact that resulted from the introduction of the infrared pen with the Wii Remote brings to light the complexity of both developing and learning to use a new interface tool. The fact that the volunteers in this experiment were given very minimal training makes it clear that much more consideration must be given when introducing a new tool to insure that its use is natural and more or less immediately clear. The particular implementation may be enhanced and made more natural in the future with a more powerful infrared pen with a wider field of emission, additional or better positioned Wii Remote cameras, and better software model for manipulation of the Wii Remote camera data. Given their minimal training and the relative complexity of the implementation, such enhancements would likely result in more adept test subjects in future experiments.

ACKNOWLEDGEMENTS

This project was supported by NIH/NINDS 1R01NS056307 and NIH/NINDS 1R21NS064534.

REFERENCES

[1]. Benner T, Wisco JJ, van der Kouwe AJ, et al. Comparison of manual and automatic section positioning of brain MR images. Radiology. 2006; 239(1):246–54. [PubMed: 16507753]

[2]. Mori S, van Zijl P. Human white matter atlas. Am J Psychiatry. 2007; 164(7):1005. [PubMed: 17606649]

[3]. Morey RA, Petty CM, Xu Y, et al. A comparison of automated segmentation and manual tracing for quantifying hippocampal and amygdala volumes. NeuroImage. 2009; 45(3):855–66. [PubMed: 19162198]

[4]. Makris N, Schlerf JE, Hodge SM, et al. MRI-based surface-assisted parcellation of human cerebellar cortex: an anatomically specified method with estimate of reliability. NeuroImage. 2005; 25(4):1146–60. [PubMed: 15850732]

[5]. Fischl B, Salat DH, Busa E, et al. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. Neuron. 2002; 33(3):341–55. [PubMed: 11832223]

[6]. Warfield SK, Zou KH, Wells WM. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. IEEE Trans Med Imaging. 2004; 23(7):903–21. [PubMed: 15250643]

[7]. Landman, B.; Wan, H.; Bogovic, J., et al. Simultaneous Truth and Performance Level Estimation with Incomplete, Over-complete, and Ancillary Data; SPIE Medical Imaging Conference; San Diego, CA. 2010;

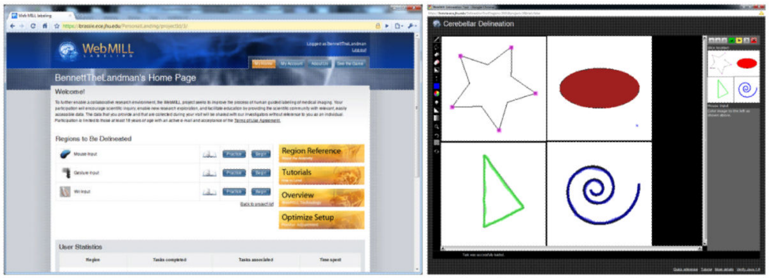

Figure 1.

WebMILL labeling system. The WebMill website provide account maintenance, progress tracking and instruction for multiple projects and tasks within projects (left) while a light weight applet provides for interaction with imaging data with multiple drawing tools and advanced zoom, undo/redo, and coloring options.

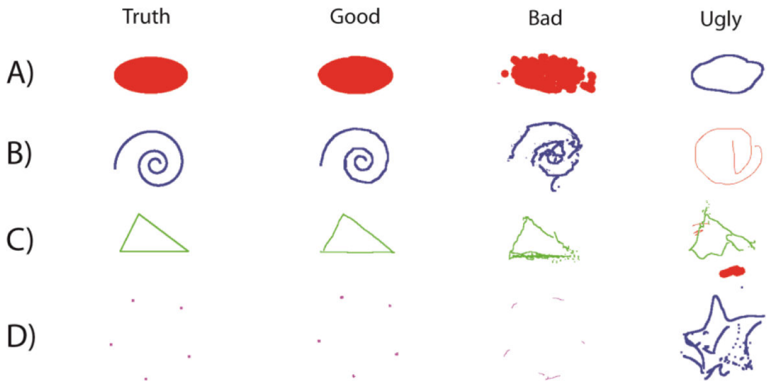

Figure 2.

Representative labeling results. For illustrative purposes, we show the range of observations divided into visually good classification (generally precise), bad classification (rules were followed but the labeled images are not visually close to the truth), and ugly classification (inconsistent with the expected ground truth). All qualities of observations were observed using all input devices.

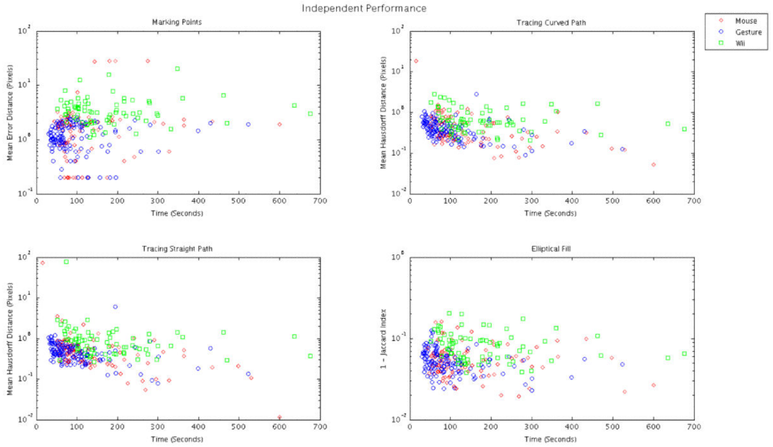

Figure 3.

Labeling error with respect to time spent independently for all sub-tasks. Each plot represents the error associated with each sub-task (i.e. points of a star (Upper Left), spiral contour (Upper Right), triangle contour (Lower Left) and elliptical fill (Lower Right) — indicated for inlay) for all individual observation with respect to the amount time spent making the observation.

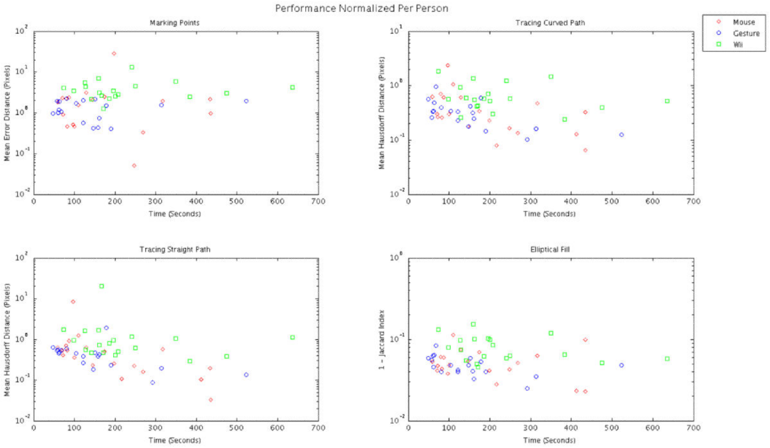

Figure 4.

Labeling error with respect to time spent normalized per person for all sub-tasks. Each plot represents the error associated with each sub-task (i.e. points of a star (Upper Left), spiral contour (Upper Right), triangle contour (Lower Left) and elliptical fill (Lower Right) — indicated for inlay) for all individual observation with respect to the amount time spent making the observation.

Table 1

Table of numerical results. This data includes the number of data samples collected for each sub-task (A), the average time and standard deviation to accomplish entire tasks using the different input methods (B), and the average error and standard deviation of each of the sub-tasks (C). Units are listed above each measurement. Note that J.I. (C) is short for Jaccard Index

	# of Data Samples				Time ± Std. Dev.	Mean Error ± Std. Dev.			
	Points	***Spiral***	***Triangle***	***Ellipse***	***Set of Four***	***Points***	***Spiral***	***Triangle***	***Ellipse***
Units	**Count**	**Count**	**Count**	**Count**	**Seconds**	**Pixels**	**Pixels**	**Pixels**	**1 – J.I.**
Mouse	101	101	103	101	133.33 ± 99.05	2.48 ± 5.29	0.630 ± 1.81	1.234 ± 6.99	0.058 ± 0.027
GesTr	130	132	135	131	97.21 ± 79.08 [p = .0024]	1.267 ± 0.643 [p = .0108]	0.434 ± 0.29 [p = .2238]	0.531 ± 0.533 [p = .2499]	0.054 ± 0.019 [p = .2400]
Wii	72	70	74	72	180.84 ± 120.40 [p = .0052]	4.019 ± 3.03 [p = .0273]	0.792 ± 0.573 [p = .4710]	1.960 ± 8.95 [p = .5426]	0.087 ± 0.397 [p < .0001]