BEYOND SCREEN CAPTURE: CREATING EFFECTIVE MULTIMEDIA WHITEBOARD LECTURES ON A TABLET PC

Brian C. Dean
Department of Computer Science
Clemson University
bcdean@cs.clemson.edu

1 Abstract

In many areas of study, researchers and educators often encounter concepts that are difficult to articulate in static, printed text. The goal of this project is to develop tablet PC software that enables the creation of rich multimedia content with which we can more effectively communicate these types of concepts. To this end, we have developed a program called LectureScribe that allows the user to create animated whiteboard lectures by simultaneously capturing voice and handwriting on a tablet PC or an electronic whiteboard. The resulting animation can be exported to a highly-compressed Macromedia Flash file for viewing over the Web on nearly any computing platform. While several screen capture software packages also allow one to achieve this same basic functionality, we argue that our program provides a much simpler, efficient, and intuitive interface for the particular task of creating animated whiteboard lectures. In this paper, we describe (sadly, in static printed text) the particular features that make LectureScribe easy to use, and we also comment on the experience of the author and his students in evaluating the overall effectiveness of using multimedia content produced with LectureScribe.

2 Problem Statement and Context

Static printed text has served as a the primary means of communicating technical content for hundreds of years. However, this format often encumbers our ability to articulate certain concepts as effectively as possible. For example, in computer science we might want to explain the operation of an algorithm that finds the best route for a robot to navigate through a maze. Since here we are describing a dynamic process that is highly geometric in nature, it is much more effective to explain such a concept in a setting where we can speak, draw pictures, gesture, and illustrate the dynamic nature of the process as it evolves over time. If we are interested in off-line creation of rich educational content with all of these features, full-motion video if a lecturer in front of a whiteboard would certainly suffice. However, video has significant drawbacks in terms of storage space, bandwidth, and the ability to edit content in a seamless fashion (for example, if we want to back up and restart the recording after misspeaking). Another approach that has gained popularity is to build a slide show with accompanying audio; while this solves the bandwidth and seamless editing problems, it does not however allow us to articulate concepts in a fully dynamic manner, as we could if we were giving an explanation in front of a whiteboard.

An attractive alternative to full-motion video and slide shows with accompanying audio is to produce what we call an *animated whiteboard lecture* by capturing the voice and dynamic handwriting of a user who is writing on a tablet PC or electronic whiteboard. The result can be compressed and streamed over even a low-bandwidth connection, and content in this format can be pieced together and edited in a fairly seamless fashion (e.g., we can record half of a lecture, then return the next day to record the second half with no noticeable discontinuity in the animation).

Screen and electronic whiteboard capture software such as Macromedia Captivate and Mimio Boardcast give us one way to create animated whiteboard lectures. However, at the present time it seems that most screen capture software is not ideally suited for this task. For example, an effective interface for recording animated whiteboard lectures must give us immediate, fine-grained control over recording and playback. During the creation of even a short segment, it is quite easy to misspeak dozens of times — as a result, the ability to quickly back up and "reset" the recording is crucially important, yet this is a somewhat painstaking procedure with today's screen capture software. In addition, screen capture software can also make it difficult to return and edit (or extend) a recording at a later date, since this often requires us to return our desktop to the precise state it was in at the time of the initial recording.

In this article, we describe a software tool, LectureScribe, that we have developed specifically to facilitate the creation of animated whiteboard lectures. We comment on the effectiveness of this tool, both in terms of the author's experience using it to create animated lectures, and also in terms of feedback from his students in classes where these animated lectures were used to supplement the class materials.

3 The LectureScribe Program

In this section we briefly summarize the main features and operation of our LectureScribe program. As we see from the screenshot in Figure 1, the program behaves roughly like a painting program with added recording and playback controls. A lecture is comprised of a series of *boards* each of which can be edited independently and arranged in any order for the final presentation. Each board is recorded by clicking the record button or by pressing 'R' and then simply writing and speaking. Recording is stopped by pressing the stop button or the 'S' key, and the result can be played back by pressing the play button or the 'P' key.

All commands in LectureScribe are accessible both on the button bar for mouse/pen input and also via keyboard shortcuts. The keyboard shortcuts, once learned, allow us to create lectures very efficiently by pressing keys and writing simultaneously. For example, during the recording process we can change drawing colors and drawing mode (pen, line, circle, and other primitives) with a single keystroke using one hand without significantly interrupting the other hand as it draws a complex figure. If we misspeak, we can back up and restart recording almost immediately — by pressing 'S' (stop), 'B' (back up), 'S' (stop), 'K' (kill from this point on), and 'R' (record).

We can draw either while the program is recording (in which case our handwriting will appear dynamically during playback) or while the program is stopped (in which case the figure we draw will appear all at one instant once the lecture reaches the current frame). This second mode can be used to create lectures that behave more like slide shows, if desired.

One drawback of moving from a full-motion video of the lecturer to only a video of his or

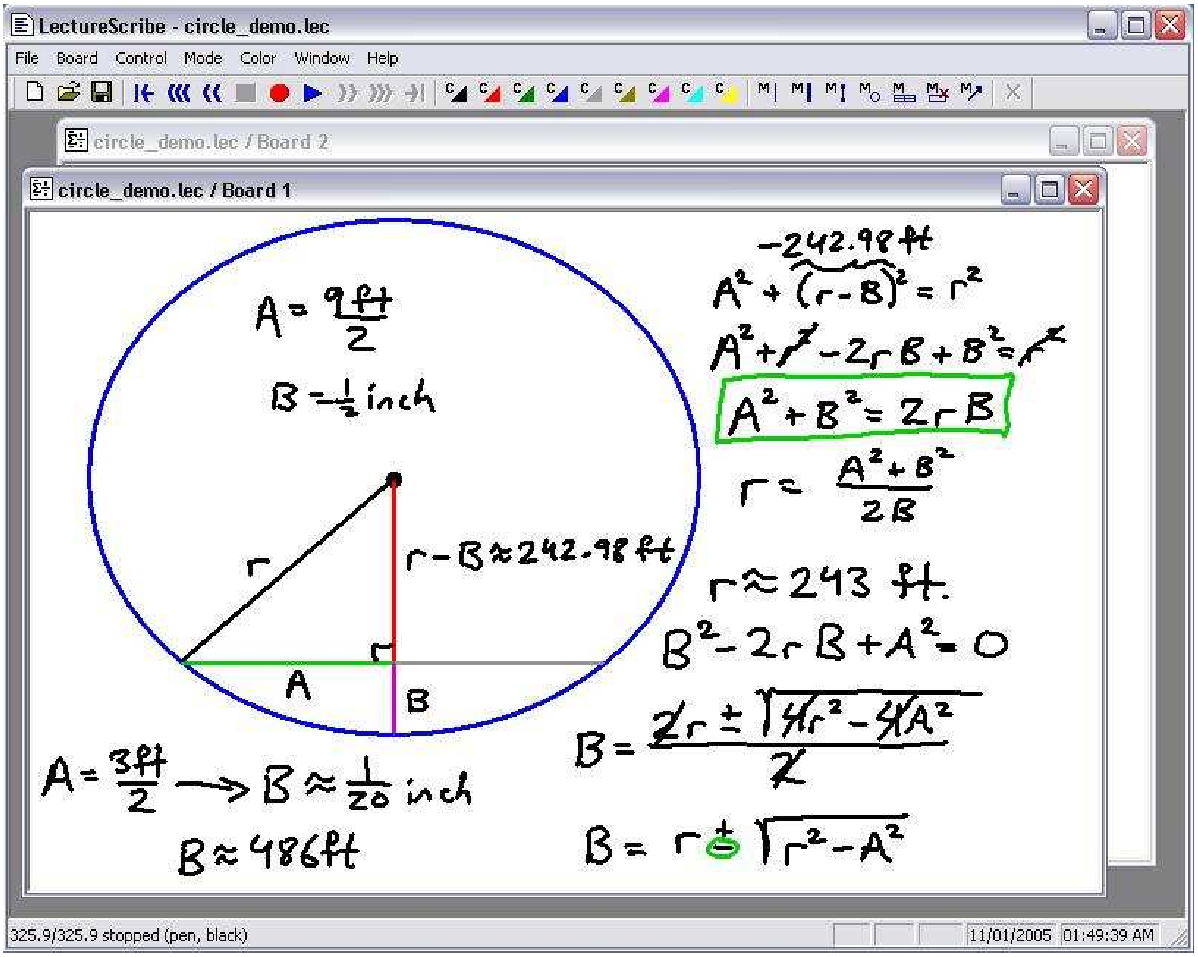
LectureScribe - circle_demo.lec
File Board Control Mode Color Window Help
circle_demo.lec / Board 2
circle_demo.lec / Board 1
$A = \frac{9\,ft}{2}$
$B = \frac{1}{2}$ inch
r
$r - B \approx 242.98$ ft
A
B
$A = \frac{3ft}{2} \longrightarrow B \approx \frac{1}{20}$ inch
$B \approx 486$ft
$A^2 + \underbrace{(r-B)^2}_{-242.98\,ft} = r^2$
$A^2 + \cancel{r^2} - 2rB + B^2 = \cancel{r^2}$
$\boxed{A^2 + B^2 = 2rB}$
$r = \frac{A^2 + B^2}{2B}$
$r \approx 243$ ft.
$B^2 - 2rB + A^2 = 0$
$B = \frac{\cancel{2}r \pm \sqrt{\cancel{4}r^2 - \cancel{4}A^2}}{\cancel{2}}$
$B = r \pm \sqrt{r^2 - A^2}$
325.9/325.9 stopped (pen, black)
11/01/2005 01:49:39 AM

Figure 1: Screenshot of the LectureScribe program. The current lecture consists of two boards, each displayed in its own window.

her handwriting on a whiteboard is that we lose the ability to make complicated hand gestures. In order to partially remedy this situation, LectureScribe allows us to switch (again, with a single keystroke) into a “gesture” drawing mode where pen strokes move around a pointer on the screen that we can use to emphasize material we have already drawn.

Images in several common formats can be imported into LectureScribe, after which we can annotate on top of the images in a dynamic fashion. For example, one could import a scanned image of a student’s assignment and dynamically mark it up while carefully explaining the subtleties behind any errors that might be present.

Once our lecture is complete, we can export it into single Macromedia Flash file, which allows for easy publication on the web or as an email attachment. No special web server software is necessary for publishing Flash content, and due to the ubiquity of the Flash player, nearly every computing platform can automatically view Flash content without the need for downloading additional software. Since Flash content is stored in a vector format, it compresses sufficiently well to allow for streaming over even a relatively low-bandwidth connection. Figure 2 shows a screen

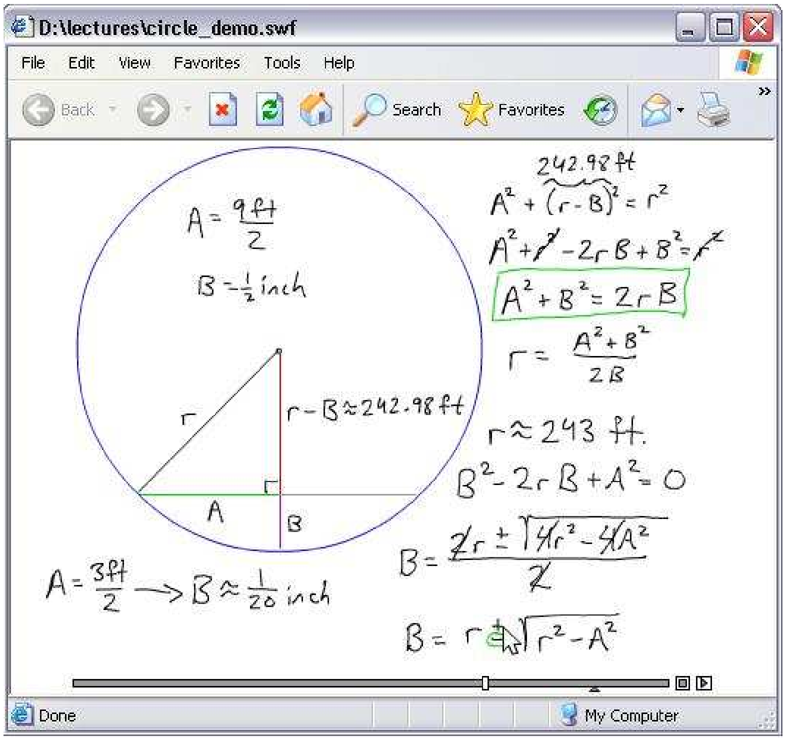
D:\lectures\circle_demo.swf
File Edit View Favorites Tools Help
Back
Search
Favorites
$A = \frac{9\text{ft}}{2}$
$B = \frac{1}{2}\text{inch}$
r
$r - B \approx 242.98\text{ft}$
A
B
$A = \frac{3\text{ft}}{2} \longrightarrow B \approx \frac{1}{20}\text{inch}$
$A^2 + \overbrace{(r-B)}^{242.98\text{ft}}{}^2 = r^2$
$A^2 + \cancel{r^2} - 2rB + B^2 = \cancel{r^2}$
$\boxed{A^2 + B^2 = 2rB}$
$r = \frac{A^2 + B^2}{2B}$
$r \approx 243\text{ ft.}$
$B^2 - 2rB + A^2 = 0$
$B = \frac{\cancel{2}r \pm \sqrt{\cancel{4}r^2 - \cancel{4}A^2}}{\cancel{2}}$
$B = r \pm \sqrt{r^2 - A^2}$
Done
My Computer

Figure 2: Our animated whiteboard lecture playing back as a Macromedia Flash file. Note the navigation bar at the bottom which is automatically added, and also note the pointer arrow being used to gesture at content already drawn.

shot of our example video during playback.

4 Evaluation

As one might imagine, it is difficult to assess the true effectiveness or educational impact of creating animated whiteboard lectures with LectureScribe in a rigorous and quantitative fashion. However, the author can share several observations based on nearly four years of experience using the program in various research and teaching activities. LectureScribe has proven to be a very useful tool for (i) recording personal research thoughts, as a sort of journal, (ii) creating quick explanations for research results to email to colleagues, (iii) creating quick explanations in response to questions received from students over email, (iv) supplementing material from a lecture when the in-class lecture either ran over time or seemed particularly confusing for students, (v) presenting homework solutions or solutions to other practice problems, and (vi) developing other supplemental material for a class. The author has employed LectureScribe for all of these purposes, and in each case he has received overwhelmingly positive feedback from his students and colleagues. The author has used LectureScribe to develop supplemental content for courses he has helped to teach, most notably undergraduate and graduate algorithms courses in the computer science departments at M.I.T. and Clemson University. Informal feedback from students in these courses suggests that these animated whiteboard lectures were very effective as a learning aid.

In general, it is very easy to develop an animated lecture very rapidly with LectureScribe, but it still takes a reasonable amount of time (say, 5 to 10 times the final length of the video) to produce what one might call a "highly polished" product. In the author's opinion, LectureScribe seems best-suited to producing short lectures (5 to 10 minutes) in an off-line, rather than in-class setting. While there is no technical reason preventing us from using the program to record live lectures on a smart whiteboard or tablet PC and projector, such lectures would likely be very boring to watch, due to the significant amount of "white space" present in an actual lecture.

Animated whiteboard lectures seem to be the most effective when they are short, self-contained explanations relating to a single topic (e.g, a short proof of a single mathematical theorem). This enables collections of short videos to be organized and linked together in a convenient manner on a web site. In addition, we can combine animations together with traditional printed lecture notes or slides to obtain a very effective "dual media" collection of educational material. High level concepts can be discussed in static text in a set of lecture notes or slides in HTML or PDF format, with embedded hyperlinks that when clicked, open a new window and start playing an animated whiteboard lecture. This dual-media approach is ideal for presenting each piece of information the format best suited for it — high-level ideas typically work best in printed text, and low-level details (e.g., complicated mathematical proofs) are often easier to explain using animated whiteboard lectures.

5 Future Work

There are quite a few exciting features we would like to add to LectureScribe in the future. The highest-priority feature we would like to add is the capability of producing lectures with subtitles that are accessible to students with hearing disabilities. Another useful feature would also be the ability to "recall" images from previous boards during the discussion on the current board — in the current version of LectureScribe, once a board finishes playing it is replaced with the next board (although the user can jump back to the old board using the navigation bar). Extended import ability would also be quite useful; for example, in a software development class or a writing class, one might want to import a segment of computer code or part of an essay for further annotation. The author also plans to investigate more effective ways to add "gestures" to whiteboard lectures, for example by highlighting regions of the screen or using real-time motion capture equipment to record the movements of one's hands.

One can envision slightly more ambitious longer-term goals for the development of future versions of LectureScribe. A particularly interesting problem to address seems to be providing an interface for developing dynamic content like an animated whiteboard lecture in a collaborative fashion. For example, person A could publish a video into which person B could insert annotations or additional dynamic content. Such a system could enable the creation of a large-scale, collaboratively-developed web repository for dynamic educational content.

The current version of LectureScribe is freely available at:

`http://www.cs.clemson.edu/˜bcdean/lscribe/`