Chen, W. et al. (Eds.) (2016). Proceedings of the 24th International Conference on Computers in Education. India: Asia-Pacific Society for Computers in Education

Photocasting: A Low-Cost Technique to Create and Disseminate Digital Lecture Notes

Manasa H. G.[a], Kavitha G. PURANIK[a], Lakshmi V. ANTIN[a] & Viraj KUMAR[b*]
[a]*Department of Information Science and Engineering, PES Institute of Technology, India*
[b]*Department of Computer Science and Engineering, PES University, India*
*viraj.kumar@pes.edu

Abstract: Instructors who wish to disseminate digital content to learners in an economical manner must consider the costs of (1) content creation (software and hardware), (2) content dissemination (data transfer costs), and (3) content access (hardware and software needed to view content). In this paper, we present a novel technique – *photocasting* – to greatly reduce these three costs in a one-to-one learning environment. Our technique has minimal requirements: a low-end smartphone for instructors and learners, and a modest additional effort by instructors after recording their digital notes. The delivery cost is primarily determined by the size of the *audio*, and is approximately 100 times smaller than the size of a traditional *video* of comparable quality. We have implemented our technique as a prototype for the Android platform, and we report initial results of a small-scale user study with three participants: a school teacher, a private tutor, and a high-school student.

Keywords: One-to-one learning, content creation, content delivery, chalk talk, photocasting.

1. Introduction

An extremely popular format for digital instruction consists of handwritten notes accompanied by audio commentary, with Khan Academy and Udacity being two large-scale exponents of this model. We propose a novel, technique – *photocasting* – to help instructors create such content in a one-to-one learning environment using low-end smartphones. Our goal is to minimize costs for (1) creating content (the additional hardware and software that the instructor must purchase), (2) disseminating content (which largely depends on the volume of data to transfer), and (3) accessing content (the additional hardware and software that the learner must purchase). A comparison of our technique with existing techniques along these cost dimensions is summarized in Table 1.

The pedagogical value of such content has been well studied, and in the context of mathematics education it has a name: *chalk talk* (Artemeva and Fox, 2010). This multimodal style of instruction is defined as “writing out a mathematical narrative on the board while talking aloud”. Fox and Artemeva (2011) argue that such instruction can be “pedagogically interactive, meaningful, and engaging”. Unlike the “chalk and talk” teaching style that is rightly criticized for being overly teacher-centric, chalk talk plays an important pedagogical function in making the process of reasoning visible (Greiffenhagen, 2014), and is more engaging than PowerPoint slides in MOOC videos (Guo, Kim and Rubin, 2014). The blend of oral and written modes appears to be more important than the precise technology used to create digital chalk talk lecture notes (Schleppegrell, 2007; Artemeva and Fox, 2011).

Table 1: A comparison of the cost components of techniques for creating chalk talk lecture notes.

Technique	Cost components		
	Creation	*Dissemination*	*Access*
Interactive whiteboards	High (hardware, software)	High/Low	Low/High
Screencasting	High (hardware, software)	High/Low	Low/High
Pencasting	Moderate (hardware, software)	Low	Low
Touchcasting	Moderate (hardware)	Low	Low
Photocasting	Low	Low	Low

2. Related work

In this section, we summarize existing techniques for creating chalk talk lecture videos, and specify the reasons why each of these techniques can be prohibitively expensive in some contexts.

Pedersen *et al*. (1993) introduced *interactive whiteboard* (IWB) technology well over two decades ago, and there is some evidence of its benefits in pedagogy (Schmid, 2008). Unfortunately, the cost of IWB technology is substantial. Hardware requirements include a touch-sensitive whiteboard, an electronic pen, a microphone and a computer to capture data. The software is also specialized, which increases the cost. Further, it can be expensive to maintain IWBs in the presence of dust, fluctuations in heat and humidity, and unreliable power. Chalk talk content created using IWBs can typically be transmitted to learners in one of two formats: an open format (usually video) or a closed format. The cost of content dissemination in a video format can be high (running to tens or even hundreds of megabytes for short lectures), but learners can access the content using standard video players. On the other hand, closed formats can be significantly more compact (which lowers dissemination costs), but learners often need to purchase special software for accessing such content. We therefore indicate the cost of dissemination and access as High/Low and Low/High respectively in Table 1.

Instructors in a one-to-one learning environment may have access to tablets, which can create chalk talk lecture content for substantially lower hardware costs than IWBs by *screencasting* (*i.e.*, capturing the instructor's voice and touch-based interaction). In some contexts, however, it may be infeasible to provide access to tablets for all instructors, even on a shared basis. Furthermore, software costs remain high with tablet-based content creation. Khan Academy's popular videos, for example, were created on a system with a computer, an $80 writing tablet, and over $200 of software for screen capture and video editing. The tradeoff in dissemination and access costs with tablets is similar to the tradeoff with IWBs, and for the same reason.

Pencasting allows instructors to create content more naturally using smart ballpoint pens. These pens are augmented with embedded cameras and can read patterns of tiny dots on special note-paper. The dots encode the position of the pen as the instructor writes, and a built-in microphone records audio. The location and audio are synchronized to recreate a video-like experience called a *pencast* (Stasko and Caron, 2010). The costs (specialized hardware, software and note-paper) are lower than for screencasts, but cannot be ignored. Interestingly, dissemination costs are very low because the dominant component of data transfer is audio. Open formats for pencasts ensure low access costs.

Touchcasting (Palmer, 2011) attempts to retain the benefits of pencasting without incurring the additional cost of procuring specialized hardware (smart pens). Instead, instructors use iPads or tablets together with touch capacitive styluses or fingers to write. The audio is synchronized with the captured handwriting, and can be disseminated in an open format similar to pencasts.

None of these techniques are feasible for instructors in cost-constrained environments where the only affordable computing devices are low-end smartphones. For such contexts, our technique relies on photographs instead of touch-based inputs, since even the cheapest smartphones have built-in digital cameras with sufficient resolution for our purpose. We therefore call this technique *photocasting*. The rest of this paper is organized as follows. Section 3 describes the process by which instructors and learners use photocasting. The technology underlying photocasting is described in Section 4, and the results of our small-scale user study are presented in Section 5. Finally, we conclude with a discussion of the limitations of our current implementation and directions for future work in Section 6.

3. Photocasting

A photocast is a sequence of one or more pages of handwritten notes, together with audio commentary. In this section, we describe our prototype Android apps: one to help instructors create photocasts, and one for learners to view photocasts. The creation process consists of four steps:

1. The instructor turns off all telephonic and data services that are likely to interrupt the recording process, and places the smartphone in a convenient location close enough to capture audio.
2. Next, the instructor launches our app, taps the **Record** button, and begins to speak and write (as shown in Figure 1). If the instructor chooses, the app will permit the screen to turn off after a

few seconds to conserve battery power. However, the instructor can tap on the screen to pause the recording as needed, and can resume by tapping **Record** again.

3. When the page of notes is complete, the instructor taps on the **Photo** button to capture a *single* photograph of the whole page. Internally, our app uses image processing techniques (described in Section 4) to identify the pixels corresponding to the written text.
4. Finally, the instructor uses our app's interface (shown in Figure 2) to synchronize the audio with a dynamic display of pixels in a manner that simulates handwriting.

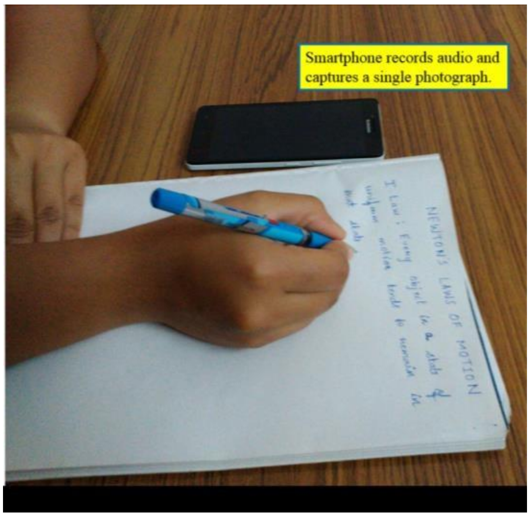

Figure 1. Creating a photocast. The smartphone captures audio while notes are written, and the built-in camera is later used to take a single photograph of the completed page.

It is important to note that the final step (4) is unnecessary for all the techniques described earlier (in Section 2), where audio is *automatically* synchronized with handwriting. In a photocast, the app will gradually reveal pixels corresponding to handwriting, but this process has to be synchronized *manually* with the instructor's audio. This step therefore represents a moderate additional cost (instructor effort) for our method. In order to keep this cost manageable, we have attempted to make our app's synchronization interface as easy to use as possible (see Figure 2). However, we acknowledge that significant improvements in usability can be achieved. We address this point further in Section 6.

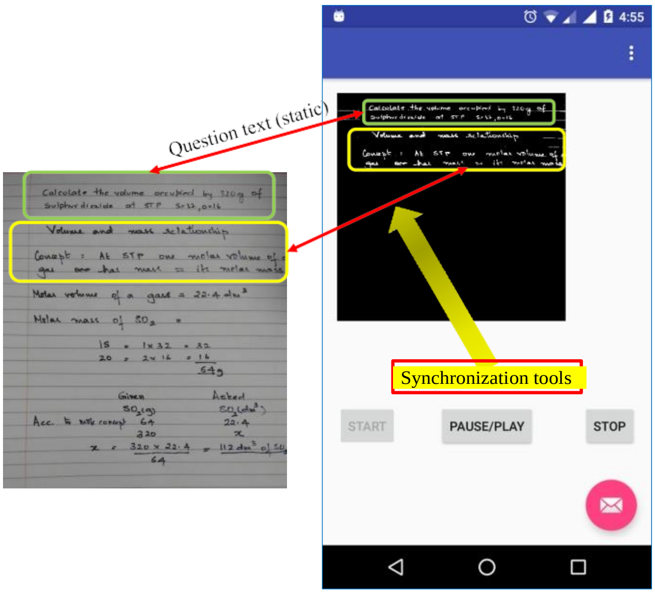

Figure 2. The photograph captured by the smartphone (left) is binarized (white handwriting on black background), and these pixels are displayed in synchrony with the audio. The instructor must perform the synchronization, using the tools provided by our app (right).

We provide **Start** and **Stop** buttons for controlling the display of pixels, and a **Pause/Play** button for controlling the audio. To begin with, both the pixel animation and the audio playback are paused. If the instructor wishes the photocast to begin with some text already written (*e.g.*, the static text of the question shown in Figure 2), she presses only the **Start** button. The pixels corresponding to her handwriting appear smoothly (left to right, line by line), and once this text is fully displayed, she presses **Pause/Play** to start the audio. To minimize the instructor's effort, our app makes a simplifying assumption that the remaining handwriting pixels should be animated at a *constant rate* to match the subsequent length of the audio. If this rate is incorrect because it is animating the pixels too fast, the instructor presses **Stop** to allow the audio to "catch up". When she presses **Start**, the animation rate is recalibrated. On the other hand, if the pixel animation rate is too slow, instructor presses **Play/Pause** and waits for the animation to reach the corresponding point in the audio. When the instructor resumes the audio by pressing **Play/Pause** once more, the animation rate of the preceding segment is recalibrated to end at the point where the audio was paused, and the subsequent animation rate is also adjusted accordingly. Our small-scale user study confirmed that this interface requires a little practice to get used to, but it is usable. The learner app consists of a simple interface to choose a photocast and play it. Learners can pause the playback and skip forwards/backwards in a manner similar to a video.

4. Technical details

In this section, we focus on the implementation details of our prototype Android app for the instructor. (The learner app to access photocasts is significantly simpler.) After the instructor has captured the photograph of her notes, we apply a series of standard image processing techniques to the raw image (or images), as shown in Figure 3. Our open-source implementation uses the OpenCV for Android SDK[1] to perform two tasks for each raw image. First, it converts the color image into an 8-bit grayscale image, and then into a binarized image using OpenCV's inverted adaptive threshold method. Next, it performs two morphological operations: erode (to enhance the contrast between the handwriting pixels and the background pixels) and d

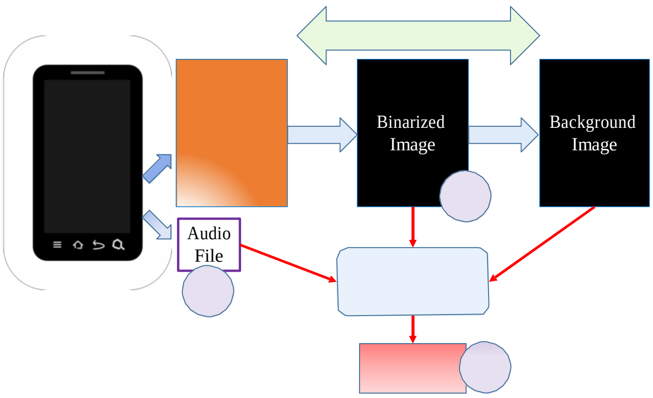

Figure 3. The raw image from the smartphone camera is processed using OpenCV libraries to obtain the binarized and background images. These, together with the audio file (also taken from the smartphone), are used to synchronize the pixel animation with the audio. The data to subsequently control the animation is stored in a JSON file. A photocast therefore consists of: (1) an audio file, (2) one or more binarized images, and (3) a control file.

By comparing the binarized and background images, our app automatically identifies chunks of (approximately) horizontal text as individual lines. Next, our app computes the total number of columns C (measured in pixels) spanned by all lines. It now examines the audio file and determines its length t (in seconds). With this data, our app determines the initial animation rate $r = C/t$. As explained in Section 3, this rate is modified when the instructor manually synchronizes the audio with the animation. The result of this synchronization is captured as a *control file* that contains a series of pairs (c, s). Each such pair (c, s) indicates that the next c columns (which may span multiple rows) should be animated

[1] Source: http://opencv.org/platforms/android.html, last accessed on June 3, 2016.

over the subsequent *s* seconds. Note that the pair (*c*, 0) indicates that the next *c* columns should be displayed *immediately*, which is useful when a block of text needs to be displayed without animation. Similarly, the pair (0, *s*) indicates that audio (but no animation) should be played for the next *s* seconds, which is useful when the instructor is orally explaining a point. We use a simple JSON format for the control file. A single photocast therefore has three components: (1) an audio file, (2) one or more binarized images, and (3) a control file (see Figure 3).

The app for learners is fairly simple. It first recomputes the background image from the binarized image (only the binarized image is transmitted, in order to minimize the dissemination cost) and splits the handwriting across lines. Next, it parses the JSON file and accordingly animates the handwriting and plays the audio.

5. User study

At this preliminary stage, we have conducted a small-scale user study with three diverse participants: a teacher who teaches Kannada (a regional Indian language) in a government school, a private tutor who coaches high school students in Chemistry, and a middle school student at a private school. All our participants were female, and their ages ranged from 15 to 56. Each of our participants had a personal smartphone running Android KitKat (or higher), whose price ranged from $75 to $190 (approximately). Each participant was given an individual 10-minute tutorial on creating photocasts, with the bulk of the time spent on explaining how to perform the synchronization step. Participants were given 5 additional minutes to synchronize a particular 90-second photocast we had created, during which time we gave as little guidance as possible. Thereafter, each participant was asked to create a photocast of her choice.

The simplest photocast was created by the school student. It took the form of a short question addressed to her teacher, and was based on a diagram of a heart value in her Biology textbook (Figure 4, left). The question asked whether the heart was to the left or right of the valve as drawn. The diagram and most of the text in this photocast was static, and the student only wrote the words “left side” and “right side” as she spoke them out. After creating the 29-second long photocast, the student took approximately 1 minute to perform the synchronization.

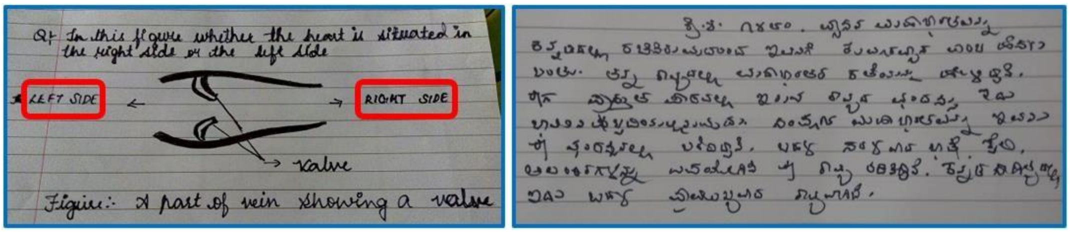

Figure 4. The raw images for the school student (left) and the school teacher (right). The bulk of the text in the left image is static (the only animated regions are highlighted in red). The right image is in Kannada (a regional Indian language).

The teacher created a somewhat longer (3 minute) photocast, whose purpose was to model a useful way to structure answers to a common type of examination problem (Figure 4, right). The handwritten text was entirely in Kannada, and we were pleased to note that there was no significant difference in visual quality between English and Kannada script when animated as a photocast. (Both scripts are read left to right, top down.) The pace at which the instructor spoke varied significantly over this short paragraph, and she included two brief explanations (where the animation needed to pause completely). During the synchronization step, the teacher struggled to pause the animation for the first explanation. After about one minute, one of us stepped in to perform this operation for her. She was able to handle the second such instance and complete the remainder of the synchronization herself.

The private tutor created a three and a half minute photocast on solving a numerical problem based on molar volumes of gases (see Figure 2). This photocast was the most complex in terms of synchronization, because the instructor paused several times for explanation, and also performed the basic mathematical operations rapidly. Nevertheless, this participant performed the synchronization on her own in good time. Table 2 summarizes the quantitative results of our user study. For these experiments, the size of each control file is 1 KB. To demonstrate the low dissemination cost of our technique when compared to screencasts, we also recorded each photocast as a video using our own smartphone. We chose the WVGA format, which provided comparable image quality to our photocast. The final column in Table 2 shows that the size of a photocast is about 1% the size of the corresponding video (a compression ratio between 80:1 and 140:1).

Table 2: Quantitative results of our user study.

User	Length (mm:ss)	Synchronizing effort (mm:ss)	*All sizes in KB*				Ratio
			Image (binarized)	**Audio (16 KHz)**	**Photocast**	**Video WVGA**	
Private tutor	03:35	05:19	50	370	421	60723	144:1
School teacher	03:13	07:32 (with assistance)	50	333	384	54620	142:1
School student	00:29	1:03	50	49	100	8315	83:1

6. Limitations and enhancements

Our present implementation has two major limitations compared to screencasting, pencasting and touchcasting. First, these latter techniques encode handwriting in a vector format that can be scaled to match the resolution of the target screen without affecting the quality of the picture. The learner interface may also support zoom in/out operations, which can be helpful when the content is viewed on a small smartphone. In contrast, in the interest of keeping costs low, our technique uses fixed-resolution binarized images, and these do not display well on very small screens. Second, we only support animation in a left to right, top to bottom manner. In contrast, all other techniques allow instructors the freedom to jump backwards and forwards during their explanations. We have experimented with allowing additional flexibility for the instructor, but our efforts thus far have increased the complexity of synchronization. We are therefore exploring ways in which we can achieve a better balance between these competing requirements.

Acknowledgements

We would like to thank all participants of our user-study for their willingness, time, effort and for their permission to use images of their work in this paper.

References

Artemeva, N. and Fox, J. (2010). Chalk Talk: A principle genre of the university mathematics classroom. In H. Graves & R. Graves (Eds.), Interdisciplinarity: Thinking and writing beyond borders.

Fox, J. and Artemeva, N. (2011). The cinematic art of teaching university mathematics: chalk talk as embodied practice. Multimodal Communication, 1, 83–103.

Greiffenhagen, C. (2014). The materiality of mathematics: presenting mathematics at the blackboard. Br. J. Sociol. 2014, 65(3):502-28.

Guo, P. J., Kim, J. and Rubin, R. (2014). How video production affects student engagement: an empirical study of MOOC videos. Proceedings of the first ACM conference on Learning @ Scale conference, Atlanta, Georgia, USA, 41-50.

Palmer, J. D. (2011). Touchcasting digital lecture notes, Journal of Computing Sciences in Colleges, 26(4), 157-163.

Pedersen, E., McCall, K., Moran, T.P., and Halasz, F.G. (1993). Tivoli: an electronic whiteboard for informal workgroup meetings. In Proceedings of the INTERCHI '93 conference on Human factors in computing systems (INTERCHI'93). IOS Press, Amsterdam, The Netherlands, 391-398.

Schleppegrell, M. J. (2007). The linguistic challenges of mathematics teaching and learning: a research review. Reading Writing Quarterly, 23, 139–159.

Schmid, E.C. (2008). Potential pedagogical benefits and drawbacks of multimedia use in the English language classroom equipped with interactive whiteboard technology. Comput. Educ. 51(4), 1553-1568.

Stasko, D. and Caron, P. G. (2010). Digital notes and lecture capture: pencasts as electronic copies of live STEM lectures and student access to digital lectures. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, 1868-1876.