

Visualizing Large-Screen Electronic Chalkboard Content on Handheld Devices

Alexander Lüning
Freie Universität Berlin
Institut für Informatik
Takustr. 9
14195 Berlin, Germany
a.luening@onlinehome.de

Gerald Friedland
Int. Computer Science Institute
1947 Center Street, Suite 600
94704-1198 Berkeley, CA, USA
fractor@icsi.berkeley.edu

Lars Knipping
Technische Universität Berlin
Fakultät II
Strasse des 17. Juni 136
10623 Berlin, Germany
knipping@math.tu-berlin.de

Raúl Rojas
Freie Universität Berlin
Institut für Informatik
Takustr. 9
14195 Berlin, Germany
rojas@inf.fu-berlin.de

Abstract

While more and more educational institutions use high-resolution, large-scale electronic displays and interactive devices as a substitute for chalkboards, we observed that an increasing number of students wants to be able to playback and view recorded lectures using their mobile devices. However, the presentation of large-scale, high-resolution content on a small-scale, low-resolution, handheld device is a challenge. Applying traditional image and video scaling methods make most of the content unreadable. This article presents a novel visualization algorithm tested with a concrete electronic chalkboard presentation system, namely the E-Chalk system. It allows viewing of large-scale electronic chalkboard content on handheld devices. In addition, this paper presents different implementations of non-interactive and interactive viewer applications that can be run and installed on different types of mobile devices.

1. Introduction

During recent years, high-resolution, large-scale displays start to be used more and more in the classroom. Usually, displays of several square meters are used as a substitute of a traditional chalkboard [7]. On the other hand, we observe a rapidly growing use of mobile devices, such as PDAs, cell phones, or handheld game consoles. In order to be easily portable, of course, the devices have to work with small displays. In order to exploit the full potential of re-

mote teaching and e-learning, it seems an obvious request to present electronic chalkboard content generated in the classroom also on handheld devices, especially since many students already made them indispensable parts of their daily lives. Also, as a result of the increasing demand for handheld devices, their computational power starts to become already comparable to usual desktop PCs from several years ago.

In the following, we present our efforts to make electronic chalkboard lectures, recorded with the E-Chalk system [2], replayable on different types of devices, including PocketPCs, the Nokia 3560 cell phone and the PlayStation Portable (PSP). Section 2 provides a short introduction to the E-Chalk system. Section 3 presents related work on scaling algorithms and other visualization techniques for handheld devices. Section 4 explains the proposed algorithm before Section 5 discusses the technical parts of its realization. Section 6 finally concludes and presents future work.

2. The E-Chalk System

The E-Chalk system was created with the idea in mind, that the chalkboard has been proven a superior teaching tool over the years. For example, its contrast is good, even in large lecture rooms. The lecturer has to think aloud while writing on the board. So the students have enough time to understand an idea or formula, ask questions and reflect on the contents of the lecture. The system consists of hardware and a software component [2].



Figure 1. A large-scale electronic chalkboard running the E-Chalk system. Photo of a datawall in a seminar room at Freie Universität Berlin.

2.1. Chalkboard Simulation

The software component simulates the basic functionality of a chalkboard and enhances it by additional features. The chalkboard simulation transforms the computer screen into a black surface where one can paint using different colors and pen widths. The board can be scrolled up and down vertically, providing the lecturer with an unlimited surface to write on. The user can also use an eraser to delete parts or all of the board content. Images can be inserted in the board and directly annotated. The system also allows the insertion of Applets that can be directly run on the board. Access to CGI scripts has been implemented as a way of interfacing web services. The board shows both textual and graphical responses. E-Chalk also provides an interface to the algebra systems Mathematica, Maple, and MuPAD.

The lecture can also be transmitted live over the Internet or played back on-demand from an archive. Remote users connect to an E-Chalk server to view everything as seen in the classroom. They can choose to receive the audio and optionally a small video of the teacher overlaid on the board [3]. The connection speed needed for a complete lecture with blackboard image, audio, and video is roughly 128 kbit/s. Without the video stream, the required connection bandwidth does not exceed 64 kbit/s. On a desktop PC, the most convenient way for users to follow a lecture is to use the Java-based playback. In this case nothing but a Java-enabled web browser is required. When viewing archived lectures, the remote user sees a control console that enables controlling the playback of the content the same as with a VCR, i. e., pause, fast-forward, and rewind.

2.2. Classroom Hardware

The E-Chalk software works with a variety of hardware that instructors can use to substitute the traditional chalkboard. Of course, the ideal electronic chalkboard would be a large, pen-sensitive screen with high display and sensor

resolution. The display must offer good contrast, so that the visual quality can be compared to a real chalkboard, e. g., it should not be required to darken the room for the lecture. A summary of possible hardware solutions is given in [1]. Of course, the idea is to conserve most of the qualitative properties of the electronic chalkboard also on the handheld device.

Figure 1 shows an example of large pen-sensitive datawall in a dedicated seminar room at Freie Universität Berlin [4]. The datawall is operated by two off-the-shelf PCs, one PC to actually work on and a second one for the pen-tracking system. A multi-head graphics card controls four screens with a total projection area of $1.15\text{ m} \times 6.13\text{ m}$ and a resolution of 4096×1024 pixels. The instructor uses a special stylus, a laser pointer with a touch-sensitive tip, to write on the board. When the pen touches the screen, the laser lights up. A vision system uses four cameras to capture a video image from the back of the datawall, in order to track the red laser spot. The laser pointer's position is mapped into the coordinates of the display system which emulates a mouse device.

3. Related Work

Many techniques exist for scaling down videos and images on a pixel-by-pixel basis [6]. Mobile phone companies offer consumer applications for scaling down movies to cell-phone format. During our experiments, we found that many of these algorithms are applicable only if the board resolution is not larger than 640×480 , the handheld resolution is at least 240×160 , and strokes are at least 3 pixel thick, which is a scaling factor 3. Otherwise, the board content becomes unreadable. Usually, a scaling of at least factor 6 is needed, for example to scale a 1280×1024 screen-resolution board to a mobile device. If only board strokes are to be scaled, simple vector graphics scaling techniques that include smoothing do a fairly good job. However, the content is still mostly unreadable because it is too small.

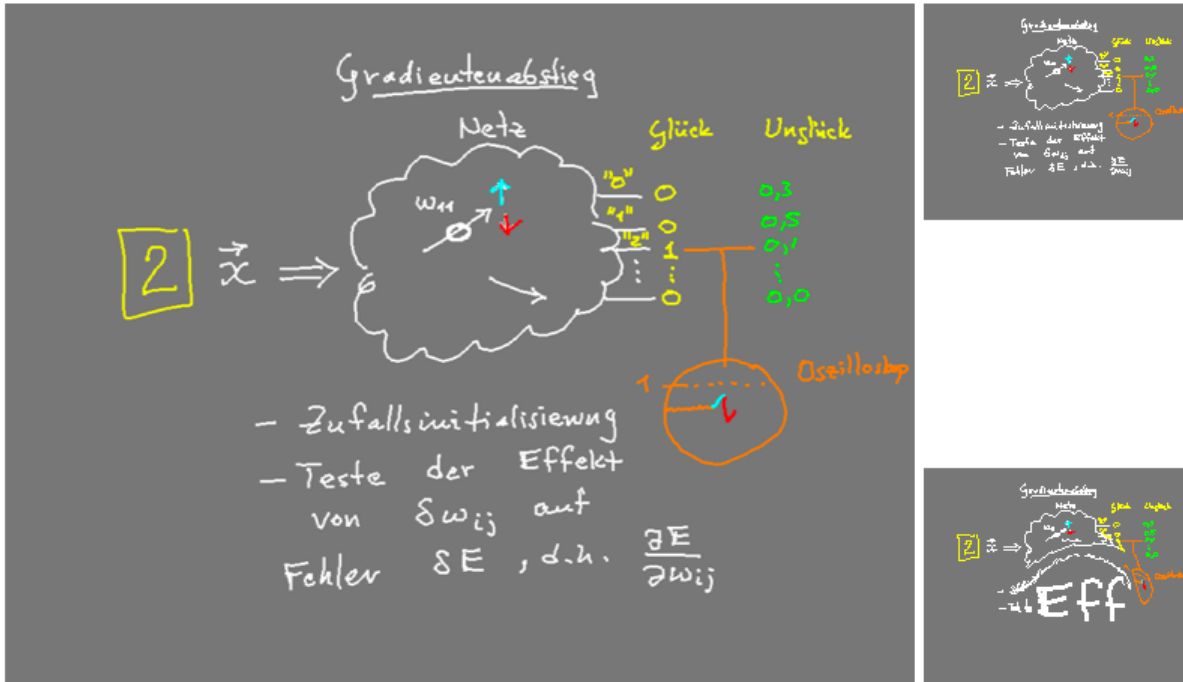


Figure 2. Left: Example chalkboard content (already scaled down by factor 2). Upper right: Same content scaled down to a Nokia 3650 using vector scaling. Lower right: Fisheye lens simulation.

An approach that has recently become popular in the HCI community [8], is the so-called fisheye view [5, 9]. This technique simulates a fisheye lens known from photography. It allows to magnify only certain parts of the screen, while a user is still able to see the entire content. However, the content at the border of the lens is radially distorted. This technique is very often used to show large scale vector graphics on small displays (for instance to visualize large trees). We found the technique can be very distracting on some devices that require the radius of the virtual lens to be very small, especially when showing tables or diagrams. The optimal radius of the lens is hard to tune automatically because it depends on the device and personal preferences. Therefore the fisheye-view replay requires interactivity on the mobile device to allow for individual lens size adjustment. In addition, the non-linear transformation to simulate the lens is quite expensive and has to be recalculated for every frame. Figure 2 provides an example of a fisheye lens simulation and a vector scaled image.

4. Proposed Visualization Algorithm

As explained in the last section, most popular scaling and visualization approaches are not suitable for the problem discussed in this paper. Fortunately, the special case of chalkboard lecture recordings offers further opportunities.

The main idea of the proposed visualization algorithm is to show only the currently interesting subregion of the board. In other words, instead of showing the entire board content, the algorithm zooms into the area that the instructor is currently working on.

The first step of the algorithm is to scale down the board content by a factor of 3 using regular vector scaling. Then, the viewport is found in the following way: A tracking window that is slightly smaller than the viewport is centered around the first activity. It is moved when the activity falls outside the current tracking window. The viewport is then adjusted accordingly. In order to avoid too sudden movements, the movement of the viewport is delayed so that a rapid movement of the activity region can be followed by the user. Figure 3 shows the idea.

As explained in Section 2, a chalkboard consists of the following components: Board strokes, images and Applets (which are replayed as animation), and text. Each of the three components has to be treated differently.

Of course, strokes are handled as described above. Images and Applets, however, may use the entire chalkboard and appear at once on the screen. Therefore, centering the viewport around the center of the picture would only provide a very limited view to the user. So when an image or an Applet is inserted into the board, the lecture replay is stopped and the image is first shown in its entirety. Then,

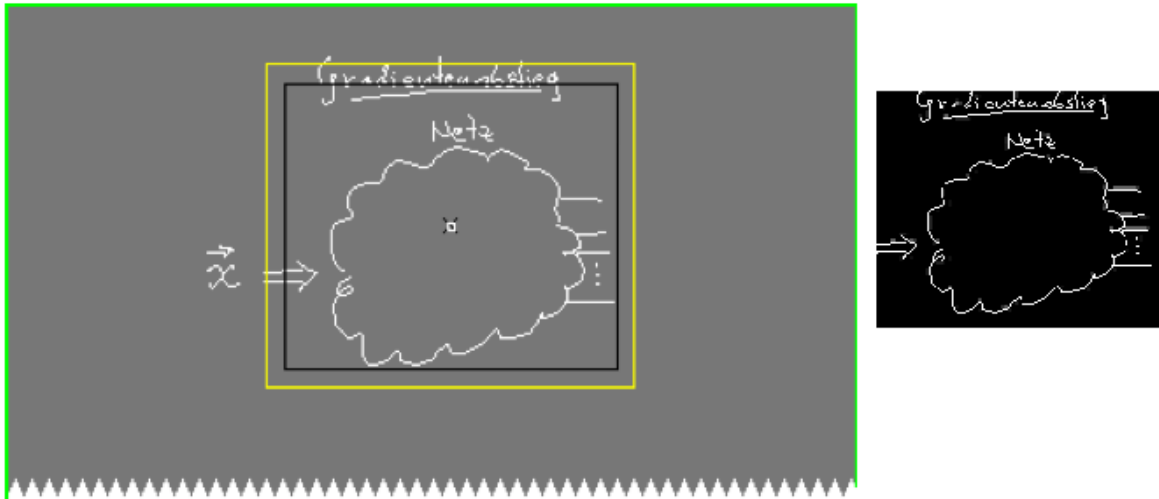


Figure 3. Left: Chalkboard picture with tracking region (black) and viewport (yellow). Right: Actual visualization on a handheld device.

the moving region scans over the picture sequentially from the top-left corner to the bottom right corner. The input of text is handled like regular writing, the viewport is centered around the currently appearing characters. When text is displayed at once, for example as the result of a call to a CGI script, the text is handled like an image or Applet.

Given a lecture recording, the algorithm described in the previous section can be easily used to create a video of the lecture. However, many handheld devices allow the execution of Java applications, usually called Midlets. This makes it possible to create packages, where a single, downloadable file contains the player and the lecture content. In addition, this allows users to manually operate the movement as well as the size of the viewport.

5. Technical Realization

The described algorithm was implemented and tested on a variety of handheld devices, including a PocketPC and a Sony Cliè, different types of Apple iPods, the Nokia 3650 and N70 mobile phones, and the Playstation Portable (PSP). Figure 4 shows some examples. Non-interactive replay of a lecture is provided by converting the board content and the audio of the lecture into either an MPEG-4 video (simple profile) or a 3GP-file. For interactive replay the lecture is not converted. The board content as well as the audio data is archived together with a player Midlet. This way, the Midlet is not only able to provide interactivity, it is also able to adapt the content to different screen sizes and color depths. A lecture recording that includes audio requires about 40–60 kbit/s. Of course, the bandwidth ultimately depends on

the content of the lecture, especially images can take more space.

6. Conclusion and Future Work

This paper presents a working solution for visualizing large-scale electronic chalkboard content on small displays of handheld devices that exploits the fact, that there is always an active region during the creation of the chalkboard content. The presented algorithm is computationally more efficient than a fisheye view because the moving zoom region can be updated incrementally. The solution has been tested and evaluated in a running system and using a set of different handhelds.

The implemented conversion system is currently integrated into the E-Chalk system, which is actively used in different universities. In the computer science department at Freie Universität Berlin alone, about 300 lecture recordings have been created since 2001 [1].

Although the presented algorithm has been realized as a usable application, many problems make the replay of lectures on small devices still a technical challenge. Unfortunately, most handheld devices do not yet support live-streaming of videos. In a Midlet, the scaling of images has to be implemented in Java because the J2ME API does not currently contain any image scaling methods. This makes the approach perceivably slow on some handhelds. Fonts on handhelds are usually not arbitrarily scalable, most operating systems provide only three font sizes.

Apart from technical issues, the proposed algorithm introduces a delay when showing images, Applets, or text



Figure 4. E-Chalk replay on different handheld devices.

output which makes live-transmissions difficult. We think, however, that live broadcasting on handheld devices will become more important. Future work also includes an easier deployment of lectures, for example via Podcasts. In addition, it would be interesting to see if a somewhat similar solution can be used for slide presentations. To spot the current active region, one could use appearing objects in animations and/or laser pointer activity.

Acknowledgements

Many others have also contributed to the development of the E-Chalk system, among them Kristian Jantz, Benjamin Jankovic, Ernesto Tapia, Christian Zick, Mary-Ann Brennan, Margarita Esponda, and Wolf-Ulrich Raffel. We want to thank the many students at both FU Berlin and TU Berlin that contributed to the success of the project by using it and providing us helpful feedback. More information on the E-Chalk project can be found at <http://www.echalk.de>.

References

- [1] G. Friedland. *Adaptive Audio and Video Processing for Electronic Chalkboard Lectures*. Ph.D. thesis, Freie Universität Berlin, Institut für Informatik, Berlin, Germany, October 2006.
- [2] G. Friedland, L. Knipping, E. Tapia, and R. Rojas. Teaching With an Intelligent Electronic Chalkboard. In *Proceedings of ACM Multimedia 2004, Workshop on Effective Telepresence*, pages 16–23, New York, New York, USA, October 2004.
- [3] G. Friedland and R. Rojas. Human-Centered Webcasting of Interactive-Whiteboard Lectures. In *Proceedings of the First IEEE International Workshop on Multimedia Technologies for E-Learning*, pages 895–900, San Diego, California, USA, December 2006. IEEE Computer Society.
- [4] G. Friedland, C. Zick, K. Jantz, L. Knipping, and R. Rojas. An Interactive Datawall for an Intelligent Classroom. In *Proceedings of the E-Lectures Workshop, Delfi Conference 2005*, Rostock, Germany, September 2005.
- [5] G. W. Furnas. Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23, Boston, Massachusetts, United States, 1986. ACM Press.
- [6] R. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2nd edition, January 2002.
- [7] L. Knipping. *An Electronic Chalkboard for Classroom and Distance Teaching*. Ph.D. thesis, Institut für Informatik, Freie Universität Berlin, Berlin, Germany, 2005.
- [8] N. F. Polys, S. Kim, and D. A. Bowman. Effects of information layout, screen size, and field of view on user performance in information-rich virtual environments. In *VRST 05: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 46–55, Monterey, CA, USA, 2005. ACM Press.
- [9] U. Rauschenbach. The rectangular fish eye view as an efficient method for the transmission and display of large images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 115–119. IEEE Press, 1999.