Chapter 9

Merging Video and Blackboard

This chapter presents an idea on how to improve the remote presentation of electronic chalkboard lectures by better utilizing the potential that computers have for multimedia processing. It discusses reasons and initial considerations that led to an enhanced approach for transmitting the non-verbal communication of the instructor in relation to the electronic chalkboard lecture, basic technical considerations, and an algorithm that implements the presented idea. Further details on the ideas and algorithms presented here can be found in [Jantz et al., 2004], [Friedland, 2004], [Friedland et al., 2005a], [Friedland et al., 2005d], and [Friedland and Rojas, 2006].

9.1 Split Attention

As discussed in Chapter 8, transmitting an additional video of the classroom context is desirable when the required connection bandwidth and storage needs can be met. For this reason, E-Chalk as well as several other lecture recording systems (see discussion in Chapter 2), transmit a supplementary video. Especially a video of the instructor conveys non-verbal information that several empiricial studies have shown to be of value for the student. There are, however, several reasons against showing a video of the lecturer next to the slides or the chalkboard visualization. The video shows the instructor together with the board content: In other words the transmitted board content is actually redundant. On low-resolution devices, the main concern is that the instructor video takes up a sigificant amount of space. The bigger the video, the better non-verbal information can be transmitted. Ultimately, the video must have the size of the board to convey every bit of information. There are also layout constraints, as the board resolution increases because electronic chalkboards become better, it gets ever more impractical to transmit the video side by side with the chalkboard content. Some window managers even arrange the board and video window in a way that the board occludes the video window. In the end, the video transmission takes up resources without the user even noticing that there is a video transmission. Even though there still might be solutions for these layout issues, a more heavily discussed topic is the issue of *split attention*.

Attention is still a topic of research and it is still an open question whether it can be split in order to attend to two or more different sources of information simultaneously (see for example [Hahn and Kramer, 1998] or [Narcisse P. Bichot and Kyle R. Cave and Harold Pashler, 1999]). The topic has been discussed by psychologists and neuroscientists for decades. Most researchers now accept that attention can be split but this usually causes cognitive overhead. One of the most important publications on the limits of human mental-performance that has strong implications on the field of man-machine interface design is still [Baars, 1988]. His *Global Workspace Theory* states that the brain has only a single locus of attention. Human beings only become conscious of information if it is selected by a central executive part of the brain. Several practical experiments that are related to the work presented here have been described in [Sweller et al., 1990] and [Chandler and Sweller., 1992].

In a typical E-Chalk lecture with instructor video (as shown by Figures 5.3 and 8.2) there are two areas of the screen competing for the viewer's attention: the video window showing the instructor and the board or slides window. [Glowalla, 2004] tracked the eye movements of students while watching a lecture recording that contains slides and an instructor video. His measurements show that a students spends about 70 percent of the time watching the instructor video and only about 20 percent of the time watching the slides. The remaining 10 percent of the eye focus was lost for activities unrelated to lecture content. When the lecture replay only consists of slides and audio, students spend about 60 percent of the time looking at the slides. Of course, there is no other spot to focus attention on in the lecture recording. The remaining 40 percent, however, were lost in distraction. However, the results may not be directly transferable to electronic chalkboard-based lecture replays because the slides consist of static images and the chalkboard window shows a dynamic replay [Mertens et al., 2006]. Motion is known to attract human attention more than static data (see for example [Kellman, 1995]), it is therefore likely that the eyes of the viewer will focus more often on the chalkboard, even when a video is presented. Nevertheless, the example shows that on a typical computer screen two areas of the screen may well be competing for attention. Furthermore, it makes sense to assume that alternating between different visual attractors causes cognitive overhead. The issue has already been discussed in [Cooper, 1990]. He provides evidence that "students presented a split source of information will need to expend a portion of their cognitive resources mentally integrating the different sources of information. This reduces the cognitive resources available for learning."

Given what has been said in Chapter 5, the introduction of the previous chapter (Section 8.1), and in this section, the following statements seem to hold:

- Replaying a traditional video of the (electronic) chalkboard lecture instead of using a vector-based representation is bandwidth inefficient, visually disadvantageous, and results in a loss of semantics.
- If bandwidth is not a bottleneck, showing a video of the instructor conveys valuable non-verbal content that has a positive effect on the learner.
- Replaying such a video in a separate window next to the chalkboard content is suboptimal because of layout constraints and cognitive issues.

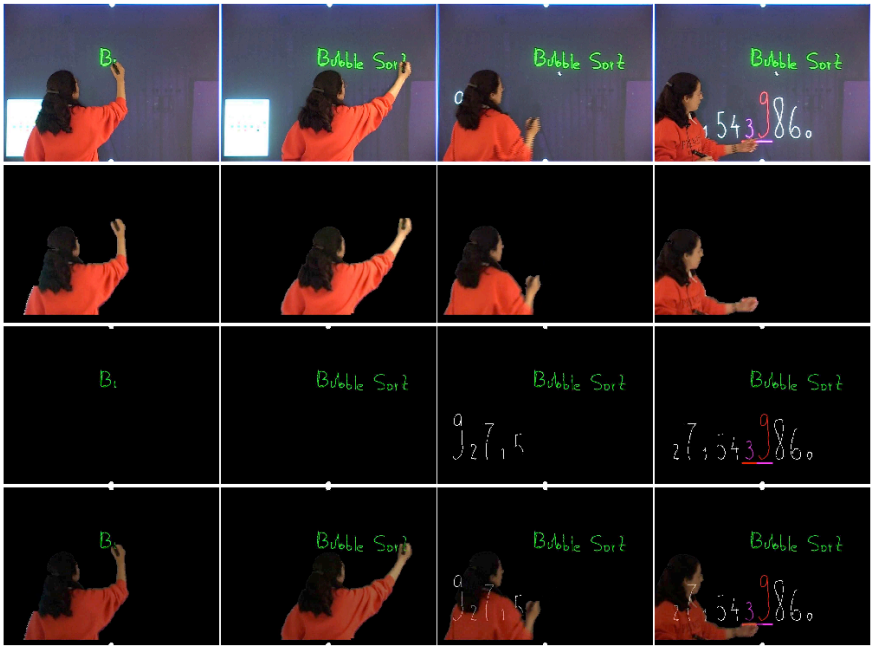

Figure 9.1: The remote viewer gets the segmented lecturer overlaid semi-transparently onto the dynamic board strokes stored as vector graphics. Upper row: original video, second row: segmented lecturer, third row board data as vector graphics. In the final fourth row, the lecturer is pasted semi-transparently on the chalkboard and played back as MPEG-4 video. The segmentation algorithm used for this picture is presented in Chapter 9.

In the following sections, an enhanced approach for transmitting the non-verbal communication of the instructor in relation to the electronic chalkboard lecture is presented. The instructor is filmed as he or she acts in front of the board by using a standard video camera and is then separated by a novel video segmentation approach that is discussed in the following chapters. The image of the instructor can then be overlaid on the board, creating the impression that the lecturer is working directly on the screen of the remote student. Figure 9.1 shows the approach. Facial expressions and gestures of the instructor then appear in direct correspondence to the board events. The superimposed lecturer helps the student to better associate the lecturer's gestures with the board content. Pasting the instructor on the board also reduces bandwidth and resolution requirements. Moreover, the image of the lecturer can be transparent. This enables the student to look through the lecturer. In the digital world, the instructor does not occlude any board content, even if he or she is standing right in front of it. In other words, the digitization of the lecture scenario solves another "layout" problem that occurs in the real world (where it is actually impossible to solve).

9.2 Related Approaches

9.2.1 Transmission of Gestures and Facial Expressions

The importance of transmitting gestures and facial expressions is not specific to remote chalkboard lecturing. In a computer-supported collaborative-work scenario people first work together on a drawing and then want to discuss it by pointing to specific details of the sketch. For this reason, several projects have begun to develop means to present gestures in their corresponding context. Two early projects of this kind were called *VideoDraw* [Tang and Minneman, 1990] and *VideoWhiteboard* [Tang and Minneman, 1991]. On both ends of the transmission a person can draw atop a monitor using whiteboard pens. The drawings together with the arms of the drawer were captured using an analog camera, so that each side sees the picture of the remote monitor overlaid on their own drawings. Polarizing filters were used to omit video feedback. The VideoWhiteboard uses the same idea, but people are able to work on a large upright frosted-glass screen and a projector is used to display the remote view. Both projects are based on analog technology without any involvement of the computer. Modern approaches include a solution presented in [Roussel, 2001] that uses chroma keying for segmenting the hands of the acting person and then overlaying it on a shared drawing workspace. In order to use chroma keying, people have to gesture in front a solid blue surface and not in front of their drawing. This has been reported to produce confusion in several situations. *LIDS* [Apperley et al., 2003] captures the image of a person working in front of a shared display with digital cameras. The image is then transformed via background subtraction into a frame containing the whiteboard strokes and a digital shadow of the person. The *VideoArms* project by [Tang et al., 2006] works with touch-sensitive surfaces and a web camera. After a short calibration, the software extracts skin colors and overlays the extracted pixels semi-transparently over the image of the display. This combined picture is then transmitted live to remote locations. The system allows multi-party communication. [Tang et al., 2004] present an evaluation of the VideoArms project. They argue that the key problem is still a technical one: "VideoArms' images were not clear and crisp enough for participants. [...] The colour segmentation technique used was not perfect, producing on-screen artifacts or holes and sometimes confusing users". In summary, the presented approaches either tried to work around object extraction, or the technical requirements for the segmentation made the systems suboptimal. It is therefore important that the lecturer extraction approach is both easily used in classroom and/or after a session, and technical requirements do not disturb the classroom lecture.

9.2.2 Segmentation

Image and video segmentation is an object of current research. Although human beings are easily able to distinguish different objects for example on a photograph, there is no generic solution that would make computers perform this task. The main obstacle here is that human vision is not well understood yet. Human visual perception takes into account not only patterns of illumination on the retina but also other senses and past experiences. Human beings are able to use context information and to fill in missing information by associating

parts of objects with already learned ones (for a detailed discussion on human vision see for example [Graham, 2001]).

All approaches that perform vision tasks on the computer – including those presented in this dissertation – are based on heuristics or on special assumptions belonging to a certain problem domain. Since many tasks that are automated with the aid of computers involve image or video processing and computer vision problems, researchers have found many spezialised tricks to achieve their goals. Current books that provide overviews of the topic include [Forsyth and Ponce, 2003] and [Bovik, 2005].

The standard technologies for overlaying foreground objects onto a given background are chroma keying (see for example [Gibbs et al., 1998]) and background subtraction (see for example [Gonzalez and Woods, 2002]). For chroma keying, an actor is filmed in front of a blue or green screen. The image is then processed by analog devices or a computer so that all blue or green pixels are set to transparent. Background subtraction works similarly: A static scene is filmed without actors once for calibration. Then the actors play normally in front of the static scene. The filmed images are then subtracted pixel by pixel from the initially calibrated scene. In the output image, regions with pixel differences near zero are defined transparent. In order to suppress noise, illumination changes, reflections of shadows, and other unwanted artifacts, several techniques have been proposed that extend the basic background subtraction approaches. Mainly, abstractions are used that substitute the pixelwise subtraction by using a classifier (see for example [Li and Leung, 2002]). Although non-parametric approaches exist, such as [Elgammal et al., 1999], per-pixel Gaussian Mixture Models (GMM) are the standard tools for modeling a relatively static background, see for example [Friedmann and Russel, 1997]. These techniques are not applicable to the given lecturer segmentation problem, because the background of the scene is neither monochromatic nor fixed. During a lecture, the instructor works on the electronic chalkboard and thus causes a steady change of the "background".

Much work has been done on *tracking* (i. e., localization) of objects for computer vision. For example in robotic soccer [Simon et al., 2001], surveillance tasks [Haritaoglu et al., 2000], or traffic applications [Beymer et al., 1997]. Most of these approaches concentrate on special features of the foreground and in these domains, real-time performance is more relevant than segmentation accuracy as long as the important features can be extracted from each video frame. Separating the foreground from more or less dynamic background is the object of current research. Many systems use complex statistical methods that require intensive calculations not possible in real time (for example [Li et al., 2003]) or use domain-specific assumptions (a typical example is [Jiang et al., 2004]). Numerous computationally intensive segmentation algorithms have also been developed in the MPEG-4 research community, for example [Chien et al., 2001]. For the task investigated here, the segmentation should be as accurate as possible. A real-time solution is needed for live transmission of lectures. [Wang and Adelson, 1994] presents a video segmentation approach that uses the optical flow to discriminate between layers of moving pixels on the basis of their direction of movement. In order to be able to track an object, the algorithm has to classify it as one layer. However, a set of pixels is grouped into a layer if they perform the same correlating movement. This makes it a useful approach for motion-based video compression but it is not perfectly suited for object extrac-

Figure 9.2: Using a stereo camera for segmentation. Top: Left and right views of the camera. Bottom left: Depth-range image, darker means farther away (white means unknown). Bottom right: Segmentation result by thresholding a certain depth, i. e., showing only pixels with depth coordinates in a certain interval. The experiment is described in the text.

tion. [Wang et al., 2003] are combining motion estimation and segmentation by intensity through a Bayesian believe network to a spatio-temporal segmentation. The result is modeled in a Markov Random Field, which is iteratively optimized to maximize a conditional probability function. The approach relies purely on intensity and movement, and is therefore capable to segment grey scale. Since the approach also groups the objects by the similarity of the movement, the same limitations as in [Wang and Adelson, 1994] apply. No details on the real time capability were given.

Research is also investigating segmentation approaches using spezialised hardware. The *Thermo-Key* project [Yasuda et al., 2004] investigated the segmentation of persons from the background using thermal cameras. The system uses the fact that the temperature of the human body is usually higher than the temperature of the surroundings, well-known, and rather constant. The presented system achieves a good illumination and texture-independent segmentation in real time. However, thermal cameras are still very expensive. The camera used in the thermo-key project cost about $ 40,000. Because the lecturer stands in front of a plane (the board), segmentation is also possible by using a 3D model of the scene. Everything closer to the camera than the board surface is considered to be the instructor. Several technologies for 3D scene analysis currently exist. 3D laser scanners, for instance, are being used with increasing frequency, for example for the conservation of historical heritage [Ogleby, 2001], special effects [18], and autonomous robots [Nüchter et al., 2003]. Usually triangulation is used to reconstruct depth information. However, the process of reconstructing the 3D model of a scene or object is computationally expensive [Bernadini et al., 2001] and far from being computable in real time.

The use of stereo cameras for the reconstruction of depth information has been thoroughly investigated (see for example [Bradski and Boult, 2001]). The different perspectives of the two human eyes lead to slight relative displacements of objects (disparities) in the two monocular views of a scene (in contrast to several animals that have two non-overlapping views, for example horses). The human visual system is not only able to merge both monocular views into a fused view of the scene, it also uses the disparities for depth-estimation. However, the correct and fast estimation of disparities is a difficult problem for computers. It is a calculation-intensive task and real-time processing requires additional hardware [Zitnick and Kanade, 2000]. Moreover, because it involves texture matching, it is affected by the same problems as texture classification methods. For example, similar or homogeneous areas are very difficult to distinguish.

Figure 9.2 shows the result of a quick segmentation test using the stereo camera "STH-MDCS2-C" by Videre Design, Inc. [65][1]. The frame rate of the camera was acceptable for low resolutions (25 frames per second at 320×240 pixels). However, at higher resolutions such as 640×480, frame rates dropped below three frames per second on a 3-GHz Pentium 4 processor even on a highly optimized version of the provided disparity estimation software with MMX support. The camera needs to be calibrated extensively before first use and the results are not suitable for the given problem. As can be seen in the picture, disparity estimation fails in similar or homogeneous regions.

Time-of-flight 3D cameras avoid the practical issues resulting from 3D imaging techniques based on triangulation or interferometry. They are currently becoming available on the market (see for example [1, 11, 13, 39]). They allow capturing of 3D data at usual video frame rates. As [Gordon et al., 1999] has already shown, range information can be used to get a better sample of the background faster. [Göktürk and Tomasi, 2004] investigate the use of 3D time of flight sensors for head tracking. They use the output of the 3D camera as input for various clustering techniques in order to obtain a robust head tracker. Since these initial results appear promising, Chapter 11 will investigate lecturer extraction using time-of-flight 3D cameras.

9.3 Setup

In E-Chalk, the principal scenario is that of an instructor using an electronic chalkboard at the front of the classroom. The camera records the instructor acting in front of the board so that just the screen showing the board content is recorded. With a zoom camera this is easily done from a non-disturbing distance (for example from the back of the classroom) and lens distortion is negligible. In the remaining chapters it will be assumed that the instructor operates using an electronic chalkboard with rear projection, such as the interactive datawall described in Section 3.2. The reason for this is that when a person acts in front of the board and a front projector is used, the board content is also projected onto the person. This makes segmentation very difficult. Furthermore, given a segmentation, the projected board artifacts disturb the appearance of the lecturer. Once set up, the camera does not require operation by a camera person. The E-Chalk Startup Wizard takes care of starting and terminating

[1] I thank Hans-Ulrich Kobialka for providing me with this camera.

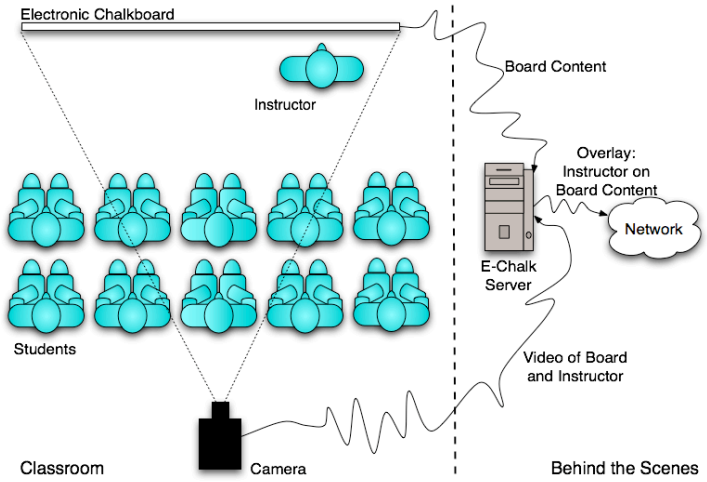

Figure 9.3: A sketch of the setup for lecturer segmentation. An electronic chalkboard is used to capture the board content and a camera records the instructor acting in front of the board.

the video recording. In order to facilitate segmentation, changes in lighting and (automatic) camera adjustments should be avoided as far as possible.

Figure 9.3 shows a sketch of the setup. The E-Chalk system is used to record chalkboard content, audio, and video. Additional SOPA nodes (see Chapter 4) take care of the segmentation. The replay of such a lecture has already been discussed in Chapter 5.

9.4 Initial Experiments

This section summarizes some of the initial experiments conducted for instructor segmentation. Even though they were never used productively for any lecture, they teach us a few facts on the nature of the given segmentation problem.

9.4.1 Simple Approaches

Very simple approaches like subtracting the board background color do not work. Figure 9.4 shows two sample camera views. Even though in both cases the background color of the board is black (RGB value $(0, 0, 0)$), the camera sees a quite different picture. Noise and reflections in particular make it impossible to threshold a certain color. Furthermore, while the instructor is working on the board, strokes and other objects appear in a different color than the board background color so that several colors have to be subtracted.

Another experiment consisted of matching the blackboard image on the screen with the picture seen by the camera and subtracting them. During lecture recording, an additional program regularly takes screenshots. The screenshots contain the board content as well as any window borders and dialogs shown on the screen. However, subtracting the screenshots from the camera view was impractical. In order to match the screen picture and the camera view, lense

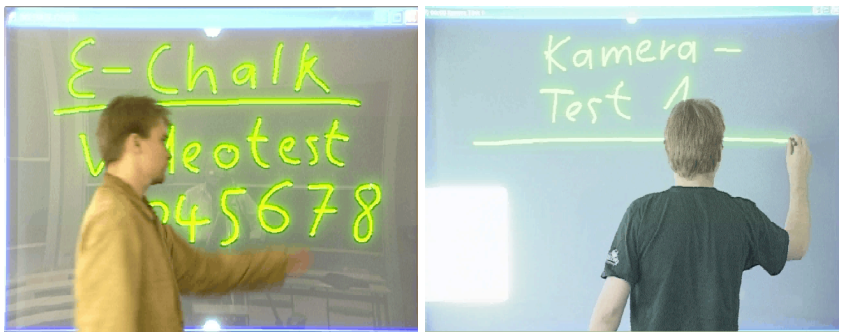

Figure 9.4: Two examples of frames captured by the video camera in the setup described in Figure 9.3. Segmentation problems include reflections of the classroom on the board, interlacing artifacts, and noise induced into the camera produced by the backlight of retroprojectors.

distortion and other geometric displacements have to be removed. This requires a calibration of the camera before each lecture. Taking screenshots with a resolution of 1024×768 pixels or higher is not possible at high frame rates. In my experiments, I was able to capture about one screenshot every second and this took almost a hundred percent of the CPU time. Furthermore, it is almost impossible to synchronize screen grabbing with the camera pictures. In a regular lecture, many things may happen during a second. After all, a matching between the colors in the camera view and the screen shots has to be found.

Instead of using screenshots, one could also require that the instructor enters the camera scene a few seconds after the start of the lecture. During this first few seconds the computer captures frames without instructor, which can later be subtracted from the images containing the instructor. The approach works very well until the instructor writes something on the board. Then, not only do board strokes appear along with the instructor, the board strokes also cause a slight illumination change, so that the background subtraction fails for large regions of the image as more and more content is put onto the board. In Figure 9.5, a worst-case example of changing illumination is illustrated. Fortunately, this worst-case scenario is rare in practice and can be dramatically reduced when using a video camera that is able to adjust to changing lighting conditions.

Geometric Assumptions

In most cases, one can assume that a human being consists of two legs, a body, two arms, and a head. Theoretically, these geometric features could be exploited to construct a model of the instructor (see for example [Remondino and Roditakis, 2003]). However, relying on many geometric assumptions or trying to extract the instructor image using geometric features is impractical. These techniques may work well for tracking, i. e., it may be possible to localize a person or an object; boundary-accurate segmentation, however, is not solved by these approaches. The instructor is seldomly seen in its entirety, because while giving a lecture in the real world, he or she will try not to occlude the chalkboard as much as possible. In most parts of the recorded instructor video, only an arm or a part of the instructor is visible. It is possible that the lecturer disappears completely or a second person might come up to the board.

Figure 9.5: Worst case example of a change of lighting conditions during a lecture. Fortunately, such bad cases are very rare and can be reduced by using a proper configuration of the video camera.

Furthermore, any feature tracked might actually be part of the board content, for example a drawing or an inserted image. Finding borders or shapes is hard since video recordings contain motion blur, interlace effects, and noise. Even simple geometric assumptions like "the instructor always begins at the bottom of the image" (i. e., the lecturer cannot fly), "a person's surface cannot grow or shrink more than a threshold from one frame to another", or "a sudden disappearance of the teacher is impossible" cannot be used practically for improving the robustness of the segmentation approach. For example, when only an arm is visible because the instructor is out of the camera's view, it is very unlikely that his or her image begins at the bottom of the frame.

9.4.2 Motion-Based Segmentation

Instead of thresholding colors or building a model of the background, an initial approach tried to extract the instructor using motion statistics. The approach uses an observation from E-Chalk's video-encoding approach. The lecturer is captured acting in front of the board and encoded using the strategy described in Chapter 8. Most of the blocks marked as opaque in the T-Frames contain the lecturer's image because from one frame to another, changes in the board content only affect a few individual blocks. Notable exceptions are the insertion of images, appearance and disappearance of dialog boxes, and scrolling of the board content. However, the set of opaque blocks almost never renders the complete instructor, and briefly obscured parts of the background have to be updated using opaque blocks, too. So the idea of the algorithm is to collect the blocks showing the instructor over several frames by comparing the non-transparent blocks appearing in each frame with a *block history table*. The history table contains a set of blocks, each associated with a counter. In the beginning the table is empty and new blocks are inserted with each counter set to 1. During the following frames, each non-transparent block is compared to the existing blocks in the history table. If a block is sufficiently similar to an already existing block in the history table, the counter is increased, otherwise the block is also inserted into the history table. After several seconds of video, the blocks in the history table with the highest counter values are assumed to belong to the instructor image. Blocks that have a low usage count are thrown

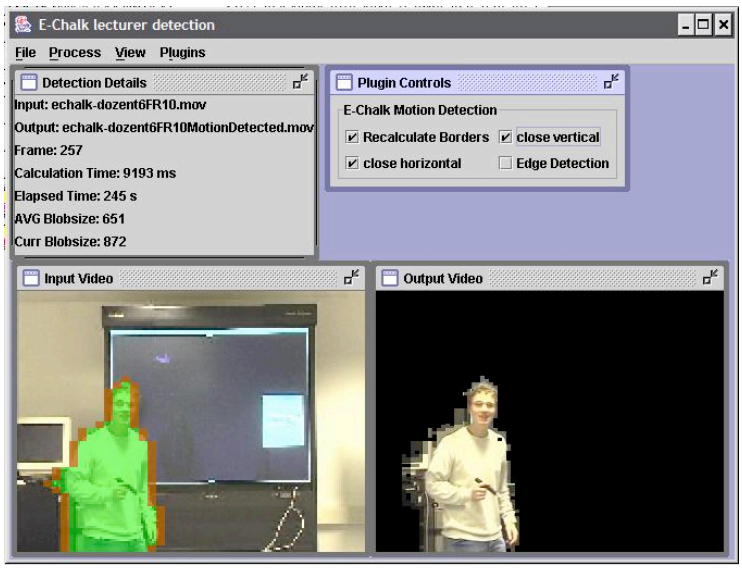

Figure 9.6: An initial instructor segmentation experiment using motion statistics. The idea is discussed in [Jantz et al., 2004].

out of the block history table. After a few seconds, the instructor is segmented by displaying those blocks in each frame that are similar to those with the highest count in the block history table. Figure 9.6 shows some results of the approach. Since long-term motion statistics are used, the approach copes pretty well with rapid changes of the board content, such as scrolling. However, when the lecturer stands still for a while, he or she disappears. Another problem are artifacts which result from working with 8×8-blocks. The biggest downside, however, is that the similarity search for each block in each frame takes too long. The experimental approach needed about 10-15 seconds per frame. The overall result is not yet robust. The details can be found in [Jantz et al., 2004].

9.4.3 A Combined Approach

Extracting the instructor using motion statistics alone is rather difficult. For this reason, a combined approach was targeted with the following experiment. The idea was to create a coarse grain cut of the foreground objects by exploiting the temporal differences between several frames and then exploiting color and color distribution information to improve the segmentation. The method is also described in [Friedland, 2004] and [Friedland et al., 2005a].

Temporal Foreground and Background Classification

The input for the classifier is a sequence of digitized YUV video frames. Each frame is subdivided into 8×8-pixel blocks. The classifier uses two main data structures:

- A foreground block buffer that is filled with any blocks that have a high chance of being part of the foreground, and

- a background buffer that contains those blocks classified as being definitely part of the background.

A block is moved into the foreground buffer if, during a sequence of n frames, the block has changed more than twice. The underlying assumption is that a background block usually changes twice when it is occluded by a foreground object and again when the view is freed. So a block that changes more than twice is an indication for a moving object (of course, background blocks close to the moving object are changed just as often as the foreground blocks replacing them). My experiments have shown that a good value is setting n to half the frame rate. A block is considered to have changed when it differs significantly from the block at the same position in the previous frame according to the Euclidean distance. The background buffer contains all blocks that have not changed during the sequence being processed, and which were never classified as foreground during later operations. Both foreground buffer and background buffer are organized as ageing FIFO queues.

Color Distribution Classification

All frames are color quantized from YUV to a fixed 256 color palette (using 4 bits for Y and 2 bits each for U and V) and are divided into 8×8-pixel blocks. For each quantized block, a color histogram is calculated. The block histograms are now classified into foreground and background by comparing each of them with block histograms of the foreground and background buffer. Because the Euclidean distance does not work very well for histograms, the *Earth Mover's Distance (EMD)* [Cohen, 1999] (and an approximation of it described in [Schindler, 2006]) was used instead.

Combining the Classifiers

The temporal classifier tends to find the borders of moving objects, while the color distribution classifier is better for surfaces. Given a frame and the results of the two classifications, any block considered foreground by at least one of the classifiers is considered foreground. The remaining blocks are a subset of the real background. For the foreground blocks, a connected component analysis is performed. The biggest blob is considered to be the instructor, and all other blocks (mostly noise or other moving objects) are put into the background buffer. Edge detection, using the Sobel Operator [Gonzalez and Woods, 1992], helps to smooth the edges of the blob, which appear ragged because of the resolution reduction to 8×8-pixel blocks. Smaller holes are filled and the corresponding block pixels are taken out of the background list. The resulting segmented video is scaled to fit the board resolution and is pasted over the board content at the receiving end of the transmission or lecture replay. Figure 9.7 shows the result already overlaid onto the board.

Although the idea seemed to be promising, the realization was far from real-time performance. At the time being, only one frame per second can be processed this way. Skin colors are difficult to extract using the approach (and also in general, compare [Zhu et al., 2004]) and block-wise operation tends to produce artifacts that are sometimes hard to retouch [Schindler, 2006]. The implementation, did not yet handle the reinitialization needed when the lecturer moves out of the video frame. Still another problem is that if the instructor

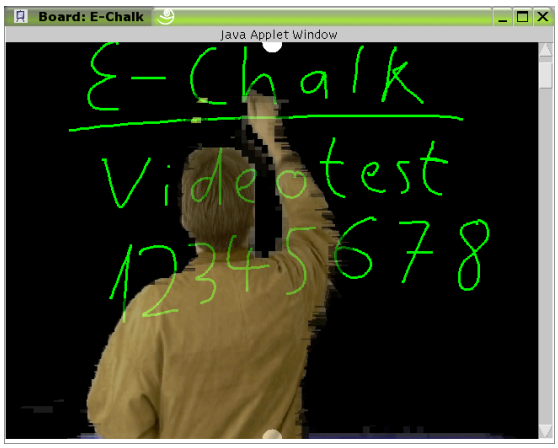

Figure 9.7: Another initial instructor segmentation experiment. For segmentation both, a temporal and a color distribution statistics was used.

points at a rapidly changing object (for example, an animation on the board screen), the two corresponding blobs could become merged.

9.4.4 Conclusion

The facts learned by the initial experiments can be summarized as follows. Tracking the instructor is insufficient, a boundary-accurate segmentation is needed. Simple color thresholding or static background subtraction models do not work because of noise and illumination changes. While the background is constantly changing, the lecturer sometimes stands still. This makes a clear distinction between foreground and background difficult using motion statistics alone. Modeling the instructor using geometric assumptions is impractical because it requires modeling all possible instructor appearances that occur in reality (cases include that only parts of the instructor are visible or several persons work on the board).

Obviously, combined approaches work better in terms of robustness. The initial experiments presented here operated on 8×8-pixel blocks. The techniques are closely connected to well-known video encoding techniques and to E-Chalk's video codec. Blocks provide a good mechanism to deal with camera-noise because they provide a good abstraction to average away outlier-colors. However, this resolution reduction produces artifacts and calculation of block similarity is computationally expensive, so another abstraction mechanism would be desirable.

Moreover, the study of literature and the experiments raise several questions. For example, it is not clear how to measure color similarity, let alone pixel block similarity. All the presented algorithms constitute heuristics that tend to include many "magic constants". Consequently, the question comes up how to measure robustness in order to make sure that a given approach actually works with many videos and not only with a few tested samples. The next section presents a segmentation approach that provides answers to many of the questions that have been raised during the initial experiments.

9.5 Robust Real-Time Instructor Extraction

As discussed in the previous section, a robust segmentation between instructor and background is hard to find using motion statistics alone. However, getting a subset of the background by looking at a short series of frames is possible. Given a subset of the background, the problem reduces to classifying the rest of the pixels into either belonging to the background or not.

The core idea behind the approach presented here is based on the notion of a color signature. A color signature models an image or part of an image by its representative colors. This abstraction technique is frequently used in different variants in image retrieval applications, where color signatures are used to compare patterns representing images, see for example [Nascimento and Chitkara, 2002, Ooi et al., 1998]. A variation of the notion of a color signature is able to solve the lecturer extraction problem and is useful for a variety of other image and video segmentation tasks. Further details on the following algorithm are also available in [Friedland et al., 2005d,Friedland et al., 2005b]. The approach presented here is based on the following assumptions: The hardware is set up as described in Section 9.3, the colors of the instructor image are overall different from those in the rest of the image, and during the first few seconds after the start of the recording, there is only one instructor and he or she moves in front of the camera. The input is a sequence of digitized YUV or RGB video frames either from a recorded video or directly from a camera. The following steps are performed:

1. Convert the pixels of each video frame to the CIELAB color space.
2. Gather samples of the background colors using motion statistics.
3. Find the representative colors of the background (i. e., build a color signature of the background).
4. Classify each pixel of a frame by measuring the distance to the color signature.
5. Apply some post-processing steps, e. g., noise reduction and biggest component search.
6. Suppress recently drawn board strokes.

The segmented instructor is then saved into E-Chalk video format. As discussed in Chapter 5, the client scales the video up to board size and replays it semi-transparently.

9.5.1 Conversion to CIELAB

The first step of the algorithm is to convert each frame to the CIELAB color space [CIE, 1978]. Using a huge amount of measurements (see [Wyszecki and Stiles, 1982]), this color space was explicitely designed as a perceptually uniform color space. It is based on the *opponent-colors theory* of color vision [Hering, 1872, Hurvich and Jameson, 1957][2]. The theory assumes that two colors cannot be both green and red or blue and yellow at the same time. As a result,

[2]Some literature even refers to Leonardo da Vinci as being the first to propose this theory [da Vinci, 1492].

Granularity	RMS Error
10	35.26
100	3.24
1000	0.32
10000	0.03

Table 9.1: CIELAB conversion approximation accuracy versus classification error. The details of the experiment are described in the text.

single values can be used to describe the red/green and the yellow/blue attributes. When a color is expressed in CIELAB, L defines lightness, a denotes the red/green value and b the yellow/blue value. In the algorithm described here, the standard observer and the D65 reference white [CIE, 1971] is used as an approximation to all possible color and lighting conditions that might appear in an image. CIELAB's perceptual color metric is still not optimal (see for example [Hill et al., 1997]) and the aforementioned assumption sometimes leads to problems. But in practice, the Euclidean distance between two colors in this space better approximates a perceptually uniform measure for color differences than in any other color space, like YUV, HSI, or RGB. Section 10.7 presents some experiments on this. Section 10.8 presents a short discussion on the limits and issues of using this color space for my purpose.

A major disadvantage of using CIELAB is the computational costs involved for the conversion from usual color spaces (usually RGB or YUV). To reduce the computational cost, I experimented with two approaches: Using a hash table to lookup already converted colors and using a lookup table filled with precalculated values to approximate the cubic roots appearing in the conversion formula. These appeared to be the efficiency bottleneck. Table 9.1 shows the classification error for different approximation granularities. The granularity defines the number of lookup values within a domain interval of size 1 for the cubic root table. The table's domain is $[0, 100]$, i. e., the number of entries is $100\times$granularity. The approximation error was measured as root mean square error over one million random pixels. The measured speed-up gained for the conversion is about a factor of 10 to 30, depending on the machine[3]. For the purpose of error reduction versus table size, a non-linear distribution of interpolation points would yield better results but then the lookup itself would be more complicated and thus slower.

Approximating CIELAB using the described approach, however, only makes sense when memory is a bottleneck. Building a hashtable, or even an entire lookup table for all 16777216 colors is not a problem on modern PC hardware.

9.5.2 Gathering Background Samples

As discussed in Section 9.4.1 it is hard to get a background image for direct subtraction. The instructor can paste images or even animations onto the board and when the instructor scrolls a page of board content upwards, the entire screen is updated. However, the instructor may also stand still, sometimes

[3]Tested on a few Windows and Linux PCs with Java Runtime Environment version 1.4 (using defaults).

Figure 9.8: Using motion statistics a sample of the background is gathered. The images show the original video (left) and known background that was reconstructed over several frames (right). The white regions constitute the unknown region.

producing less pixel changes than the background noise. The idea is thus to extract only a representative subset of the background that does not contain any foreground for further processing. The following approach assumes a non-static instructor over an inital period of a few seconds.

To distinguish noise from real movements, I use the following simple but general model. Given two measurements m_1 and m_2 of the same object, with each measurement having a maximum deviation e from the real world due to noise or other factors, it is clear that the maximum possible deviation between m_1 and m_2 is $2e$. Given several consecutive frames, e is estimated to find out which pixels changed due to noise and which pixels changed due to real movement. To achieve this, the color changes of each pixel (x, y) is recorded over a certain number of frames $t(x, y)$, called the *recording period*. It is assumed that in this interval, the minimal change should be caused only by noise. The image data is continuously evaluated. The frame is divided into 16 equally-sized regions, and changes are accumulated in each region. Under the assumption that at least one of these regions was not touched by any foreground object (the instructor is unlikely to cover the entire camera region), $2e$ is estimated to be the maximum variation of the region with the minimal sum. I then join all pixels of the current frame with the background sample that during the recording period $t(x, y)$ did not change more than my estimated $2e$. The recording period $t(x, y)$ is initialized within one second and is continuously increased for pixels that are seldom classified as background, to avoid adding a still-standing foreground object to the background buffer. In my experiments, it took a few seconds until enough pixels could be collected to form a representative subset of the

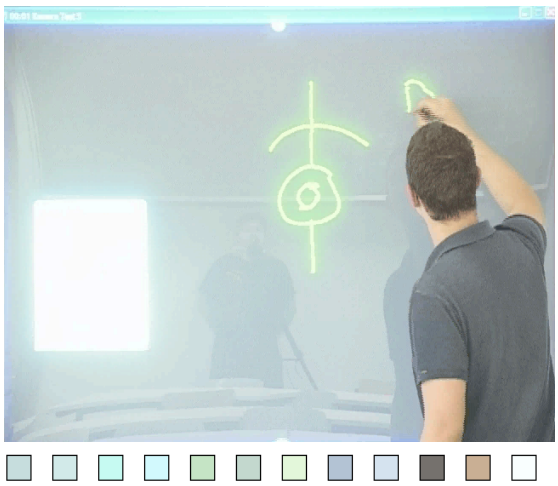

Figure 9.9: Original picture (above) and a corresponding color signature representing the entire image (below). For visualization purposes, the color signature was generated using very rough limits so that it contains only a few representative colors.

background. I call this time period the *initialization phase*. The background sample buffer is organized as an aging FIFO queue. Figure 9.8 shows typical background samples after the initialization phase.

The background sample is fed into the clustering method described in the following section. Once built up, the clustering is only updated when more than a quarter of the underlying background sample has changed. However, constant updating is still needed in order to be able to react to changing lighting conditions. The algorithm needs about one to two seconds to recover from the example illustrated in Figure 9.5.

9.5.3 Building a Model of the Background

The idea behind color signatures is to provide a means for abstraction that sorts out individual outliers caused by noise and small error. A color signature is a set of representative colors, not necessarily a subset of the input colors. While the set of background samples from Section 9.5.2 typically consists of a few hundreds of thousands of colors, the following clustering reduces the background sample to its representative colors, usually about a few hundreds. The known background sample is clustered into equally-sized clusters because in CIELAB space specifying a cluster size means to specify a certain perceptual accuracy. To do this efficiently, I use the modified two-stage k-d tree [Bentley, 1975] algorithm described in [Rubner et al., 2000], where the splitting rule is to simply divide the given interval into two equally-sized subintervals (instead of splitting the sample set at its median). In the first phase, approximate clusters are found by building up the tree and stopping when an interval at a node has become smaller than the allowed cluster diameter. At this point, clusters may be split into several nodes. In the second stage of the algorithm, nodes that belong to several clusters are recombined. To do this, another k-d tree clustering is performed using just the

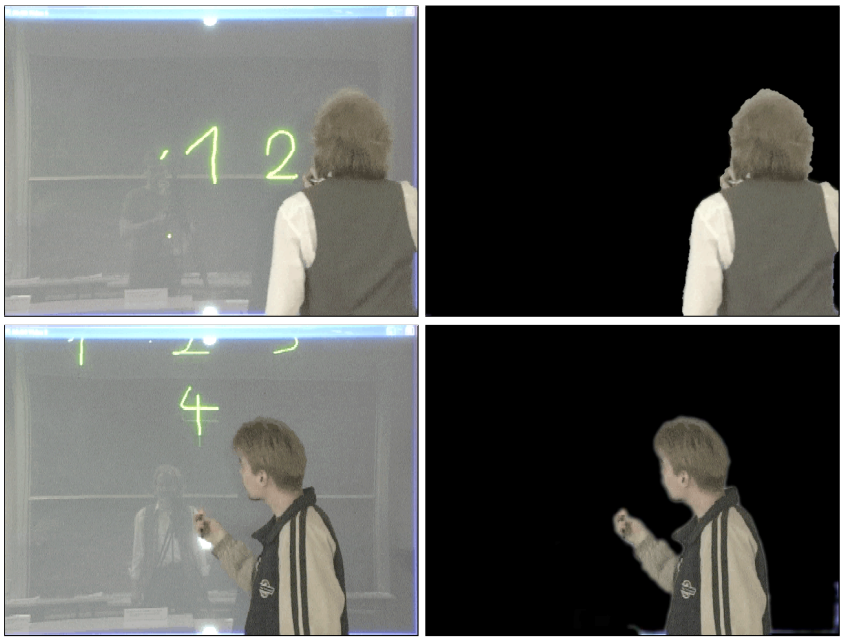

Figure 9.10: Two examples of color segmented instructor videos. Original frames are shown on the left, segmented frames are shown on the right. The frame below shows an instructor scrolling the board, which requires an update of many background samples.

cluster centroids from the first phase. I use different cluster sizes for the L, a, and b axes. The values can be set by the user according to the perceived color diversity in each of the axes. The default is 0.64 for L, 1.28 for a, and 2.56 for the b axis. For further abstraction, clusters that contain less than 0.1 % of the pixels of the entire background sample are removed. Section 10.7 explains the determination of the constants.

The k-d tree is explicitly built and the interval boundaries are stored in the nodes. Given a certain pixel, all that has to be done is to traverse the tree to find out whether it belongs to one of the known background clusters or not. Figure 9.9 shows a sample color signature.

9.5.4 Postprocessing

The pure foreground/background classification based on the color signature will usually select some individual pixels in the background with a foreground color and vice versa, resulting in tiny holes in the foreground object. The wrongly classified background pixels are eliminated by a standard “erode” filter operation while the tiny holes are filled by a standard “dilate” operation. A standard Gaussian noise filter smoothing reduces the amount of jagged edges. A biggest connected component search is to be performed. The biggest connected component is considered to be the instructor, and all other connected components (mostly noise and other moving or newly introduced objects) are eliminated from the output image. Figure 9.10 shows two sample frames of a video where the instructor has been extracted as described here.

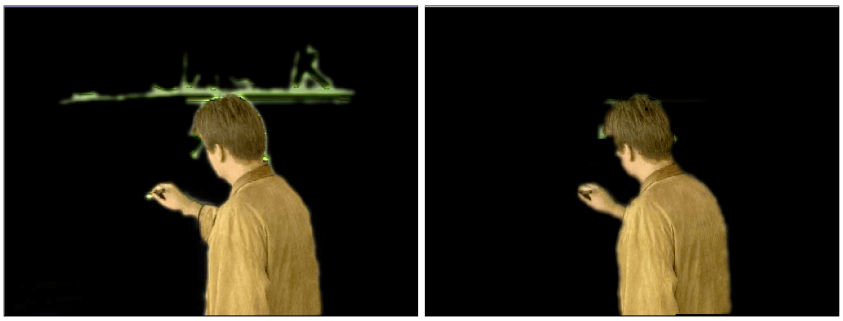

Figure 9.11: Board drawings that are connected to the instructor are often considered foreground by the classification. An additional board stroke suppression eliminates these artifacts. Left picture: The result of the color signature classification. Right picture: After applying a postprocessing step to eliminate board strokes.

9.5.5 Board Stroke Suppression

As described in Section 9.5.2, the background model is built using a statistics over several frames. Recently inserted board content is therefore not part of it. For example, when a macro is used on the board (as described in Chapter 3), a huge amount of new board content is shown on the board in a short time. With the connected component analysis performed for the pixels classified as foreground, most of the unconnected strokes and other blackboard content has already been eliminated. In order to suppress strokes just drawn by the lecturer, all colors from the board system's color palette are inserted as cluster centroids to the k-d tree. However, as the real appearance of the writing varies with projection screen, camera settings, and illumination, not all of the board activities can be suppressed. Additionally, strokes are surrounded by regions of noise that make them appear to be foreground. In order to suppress most of those thinner objects, i. e., objects that only expand a few pixels in the X and/or the Y-dimension, are eliminated using an erode operation. Fortunately, a few remaining board strokes are not very disturbing because the segmented video is later overlaid onto the board drawings anyway. Figure 9.11 compares two segmented frames with and without board stroke suppression.

9.6 Example Results

The resulting segmented video is scaled to fit the board resolution (usually 1024×768) and is pasted over the board content at the receiving end. Several examples of lectures that contain an extracted and overlaid instructor can be seen in different chapters of this document, including Figures 9.1, 9.12 and 9.13.

Reflections on the board display are mostly classified as background and small moving objects rarely make up the biggest connected component. Thresholding the minimum size of the biggest component improves the stability when the instructor leaves the camera's field of view. For the background reconstruction process to collect representative background pixels, it is not necessary to record a few seconds without the instructor. The only requirement is that for the first few seconds of initialization, the lecturer keeps moving and does

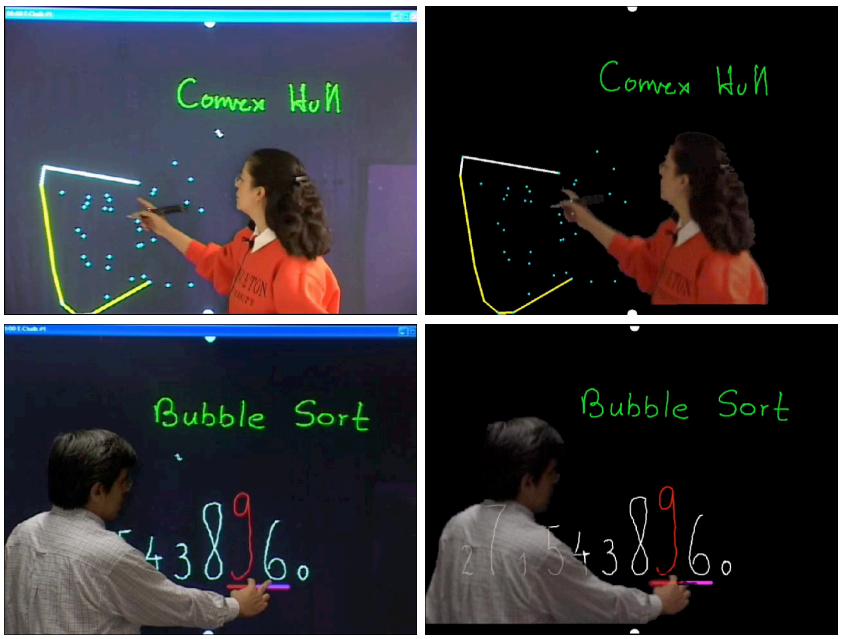

Figure 9.12: The instructor is extracted from the original video (left) and pasted semi-transparently over the vector-based board content (right).

not occlude background objects that differ significantly from those in the other background regions.

The performance of the algorithm depends on the complexity of the background and on how often it has to be updated. Usually the current Java-based prototype implementation processes a 640×480 video at 25 frames per second after the initialization phase. This includes a preview window and the E-Chalk video compression (on a PC with a 3-GHz Intel Pentium 4).

As the algorithm focuses on the background, it provides rotation and scaling-invariant tracking of the biggest moving object. The tracking still works when the instructor turns around or when he leaves the scene and a student comes up to work on the board. Once initialized, the instructor does not disappear, even if he or she stands absolutely still for several seconds (which is actually very unusual).

9.7 Limits of the Approach

The most critical drawback of the presented approach is the requirement that the instructor moves at least during the initialization phase. During our experimental recordings, we did not find this to be impractical. However, it requires some knowledge and is therefore prone to usage errors. The quality of the segmentation is suboptimal if the instructor does not appear in the the picture during the first few frames or does not move at all. Too much camera noise during the initilization phase is most often the cause for a bad segmentation result. Another problem is that if the instructor points at a rapidly changing object (for example, an animation on the board screen) of a similar color structure, the

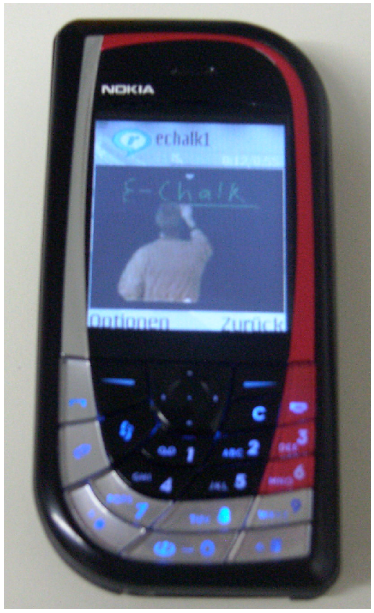

Figure 9.13: A 90-minute lecture containing dynamic board content, audio, and superimposed lecture can be played back on mobile phones. The lecture requires about 40 MB of storage space.

instructor and the animation might both be classified as foreground. If they are connected somehow, the two corresponding components could become merged and displayed as the biggest single component.

Although the instructor videos are mostly well-separable by color, the approach fails when parts of the instructor are very similar to the background. When the instructor wears a white shirt, for example, the segmentation sometimes fails because E-Chalk's toolbox often also appears white to the camera (compare Figure 9.4). One of the ideas was to improve the situation by combining the color segmentation approach with edge detection algorithms. However, the experiments failed because the videos taken in front of a rear projection are often very noisy, which results in many wrongly detected edges. Interlace effects and the internal color quantization of the camera can produce further false edges. On the other hand, when the instructor moves, the edges between the person and the board are blurred. At the edges and in high textured regions, *spill colors* can sometimes be observed. Spill colors arise when pixels contain a mix of the colors between foreground and background. This happens especially at borders of objects or highly structured regions when a pixel contains parts of an object as well as parts of the background. Removing spill-colors requires sub-pixel-accurate segmentation.

9.8 Conclusion

Using the presented segmentation approach, a solution to the split attention problem has been implemented. This improves the quality of the lecture replay at the receiving end. The lecturer is cut out of the video stream and pasted

onto the vector-based dynamic board image. The superimposed lecturer helps the student to better associate the lecturer's gestures with the board contents and conveys facial expressions. Pasting the instructor on the board also reduces space and resolution requirements. This makes it possible to replay an E-Chalk lecture on a mobile device even if it includes a video of the lecturer (see Figure 9.13). A lecture containing board, overlaid instructor, and audio can be played back on a handheld device at 64 kbit/s.

The next chapter generalizes the instructor video segmentation to a general framework that can be applied to various segmentation tasks. The generalization is able to solve several of the problems discussed in the previous section. The next chapter also presents a more detailed discussion on the limits and drawbacks of the color signature segmentation approach and presents an evaluation of the performance and robustness of the approach.