Chapter 5

Audio and Video Servers

5.1 The WWR Audio Server

The audio subsystem used for E-Chalk is a pure Java successor of the World Wide Radio (WWR) system [FL98] [109], initially named WWR2 [Man99], with the change to a new audio format renamed WWR3. The system is a TCP/IP-based audio streaming system, that uses lossy compression to achieve interruption-free transmission over small-bandwidth Internet connections. In order to guarantee continuous playing, sound data have to be buffered, creating a small delay between recording and playback, see Section 7.3.

5.1.1 Server Architecture

The WWR3 streaming server is modeled as a flow graph, see Figure 5.1 for an example. The graph consists of five basic types of nodes: Sources, Targets, Forks, Mixers, and Pipes. Technically, any component that wants to live inside the graph becomes a node by inheriting one of these five superclasses.[1] The graph that connects these components to a directed graph is specified by an XML file. Nodes are described via general properties, for example, to make the system select a certain audio codec that compresses down to a certain bandwidth. The system then searches for an appropriate codec, first locally and then in the Internet.

The structure of the graph can be changed and the nodes can be updated while the system is running. All data are synchronized automatically. This allows clients to trigger the server to be updated on the fly to match their requirements, instead of forcing the client to load an updated module. For example, nodes exists for both WWR and Windows Media Player. The system can be configured to load the appropriate modules when a client connects with either Windows Media Player or with the Java-based client for a live connection.

The nodes of the flow graph are realized as components of the *SOPA* (self-organizing processing and streaming architecture) framework [FP04]. SOPA is built on top of the *OSGi* [71] (open services gateway initiative) standard, a mechanism originally coming from the field of ubiquitous computing. It specifies

[1] This task requires a developer to overwrite up to ten methods for building a wrapper to a streaming service one wants to integrate.

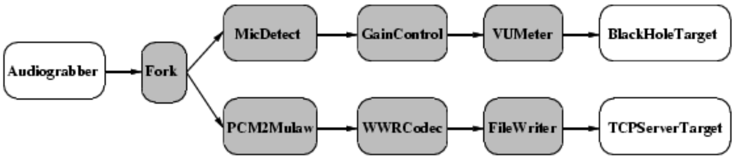


Figure 5.1: A simple audio streaming server graph.

how to load, update, and delete software components from the Internet while a system is running [OSG02, HC04a]. SOPA uses the *Oscar* OSGi implementation [HC04b] [70].

5.1.2 Audio Codecs

WWR2 Codecs

Current browsers often only have a Java environment of Java version 1.1 preinstalled. In such environments, Java Applets cannot play back sound with a sampling frequency higher than 8999 Hz [38]. Hence, the basic input format chosen for WWR2 was 8 kHz, 8 bit μ-law mono (as defined in [Int88]), the format used for digital telephone lines. It requires a connection speed of 64 kbps. Several codecs for different bandwidth, CPU speeds, and audio-quality levels were integrated into WWR2 by modifying older compression standards.

WWR2 contains a simple and fast 50 kbps codec, which uses no compression but the Java's built-in *gzip* algorithm. To achieve better compression, WWR2 also contains the 4-bit version of the μ-law codec, which is adapted from [Int88]. This codec, together with gzip, compresses down to 20 kbps. To achieve a good trade-off between sound quality, compression, and execution speed, the *ITU ADPCM* [Int90] was modified. The result were 4-bit, 3-bit and 2-bit modified ADPCM codecs that, combined with gzip, give an effective average compression of 30 kbps, 22 kbps, and 15 kbps.

WWR3 Codecs

In order to improve the sound experience for remote users with Java plug-ins installed in browsers supporting 16 kHz (Java 1.3 or later), the WWR3 audio format uses 16 bit, 16 kHz linear mono audio stream. This can also be played directly, if the browser uses Java 1.3 or later. In Java environments that only support 1.2 or earlier, the audio client converts the data to 8 bit, 8 kHz μ-law to replay them. By this means, all Java-enabled browsers can play back the stream while users with a more recent Java version can enjoy the audio at a higher quality.

The higher sampling rate increases the amount of data. WWR3 codecs have been implemented for 32 kbps, 48 kbps, 64 kbps, and 128 kbps, with the numbers designating the maximum bandwidths needed. However, the format change was accompanied with the introduction into the recording system of filtering techniques[2] that drastically reduced the amount of recorded noise. In

[2] See Section 5.2 for a description.

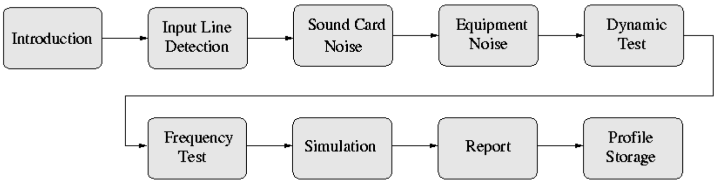


Figure 5.2: The steps of the audio profile wizard.

practice, the required bandwidth is much lower, as the cleaner signal compresses significantly better, see part on filtering techniques in Section 5.2.4.

5.2 Smart Audio Recording

As described in Section 8.4.2, the audio quality in replay was the most important weakness of the E-Chalk system according to user evaluations. This caused the change from WWR2 to WWR 3 format, but a much greater impact can be achieved by ensuring the quality of the recorded signal before encoding.

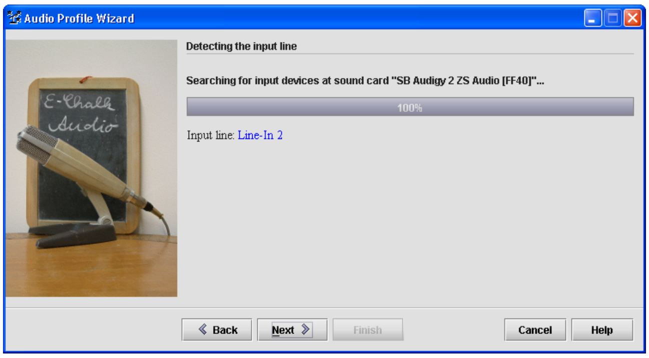
Although often advertised, most current audio recording systems cannot yield professional quality recordings just by plugging a microphone into a sound card and starting the lecture. Good-quality recordings require full-time technicians to set up and monitor the signals, or expensive professional-grade equipment, or both. The idea was to replace the technicians and the specialized hardware by self-controlling software components as far as possible [FKT04]. It is important that the system relieve the user of the technical setup details. Currently, the audio technology is not easily usable for a layperson:

> *[...] generally the networking and video setup was fairly simple, but audio setup and operation was cumbersome. We had continual problems with audio levels, microphone placement, mixer setups, feedback etc. Following the audio path from a microphone to remote speaker it turns out that there were many (eight or nine) points where audio gain could be adjusted. This, and the fact that the microphones were frequently mispositioned, caused many difficulties in establishing stable and comfortable audio levels.* [Tsi99]

5.2.1 Sources of Interference

Unfortunately, there are many potential audio distortion sources in lecture halls and classrooms. Those with the greatest impact on the recording will be mentioned here. For a more detailed discussion of these problems see for example [Dic97, Kat02].

Any room is filled with multiple sources of noise: Students are murmuring, doors slam, cellular phones ring. Reverberation effects depend on the geometry of the room and on the amount of occupied seats. Professional speakers are trained to keep their voice up at a constant level, but lecturers rarely do so. When the audience get less quiet at the end of a lecture, teachers unconsciously raise their voice with negative effects to the recording quality if the signal is



Audio Profile Wizard

Detecting the input line

Searching for input devices at sound card "SB Audigy 2 ZS Audio [FF40]"...

100%

Input line: Line-In 2

Figure 5.3: Soundcard ports are scanned for input devices.

not leveled out. Even movements of the lecturer can create noise. Coughs and sneezes, both of the audience and the speaker, result in irritating sounds. Additional noise can be introduced by the sound equipment: Hard disks and fans in the recording computer produce noise, cables can cause electromagnetic interference that results in noise or humming. Feedback loops can also be a problem if the voice is amplified for the audience.

The lecturer's attention is entirely focused on the presentation so that technical problems like just forgetting to switch the microphone on are easily overlooked. Weak batteries in the microphone cause a drastic reduction in the signal-to-noise ratio, often without the speaker noticing. Many people also have problems with the operating system's mixer. It differs from sound card to sound card, and from operating system to operating system and usually has many knobs and sliders with potentially wrong settings. Selecting the right port and adjusting mixer settings can take even experienced users minutes.

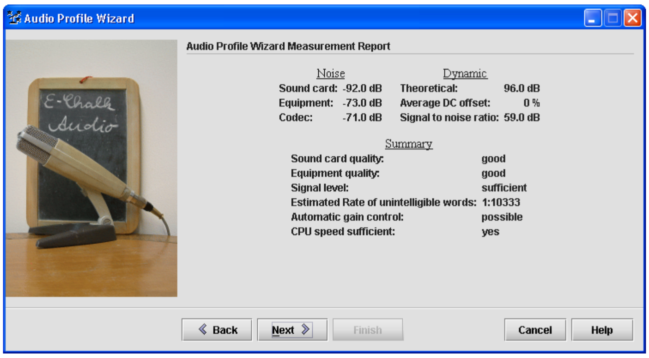
Another subject is equipment quality. Some sound cards cannot deliver high-fidelity audio recordings. In fact, most sound cards focus on sound playback but not on sound recording. Games and multimedia playback are their most important applications. On-board sound cards, especially those in laptops, often have very limited recording capabilities.

The quality loss introduced by modern software codecs is perceptually negligible compared to the problems described above. Improving audio recording for lectures held in lecture halls means first and foremost improving the quality of the input signal before it is processed by the codec. Having audio technicians and professional-grade hardware dealing with these problems is no feasible solution for our scenarios due to their high costs.

5.2.2 Enhancing Recordings

The approach taken by E-Chalk focuses on the special case of lecture recording. The system relies on the lecturer using some kind of directional microphone or a headset. Both eliminate the influence of room geometry and of cocktail-party noise [Hay03]. Headsets provide good quality but they may restrict the mobility of the speaker.

A lecture recording system like E-Chalk has the advantage that information



Audio Profile Wizard Measurement Report

Noise		Dynamic	
Sound card:	-92.0 dB	Theoretical:	96.0 dB
Equipment:	-73.0 dB	Average DC offset:	0 %
Codec:	-71.0 dB	Signal to noise ratio:	59.0 dB

Summary	
Sound card quality:	good
Equipment quality:	good
Signal level:	sufficient
Estimated Rate of unintelligible words:	1:10333
Automatic gain control:	possible
CPU speed sufficient:	yes

Figure 5.4: A report gives a rough estimation of the quality of the equipment. Speech intelligibility is calculated according to IEC 268.

about speaker and equipment are accessible in advance. Using this fact divides the approach into two parts:

- An expert system presented via a GUI wizard guides the user through a systematic setup and analyzes the recording equipment and the speaker's voice. This information is stored for the later recording phase. The wizard assists users in assessing the quality of the audio equipment and makes them aware of the equipment's influence on the recording.

- During recording, filters, hardware monitors, and automatic gain control work with the information collected by the expert system. Classic recording-studio equipment like graphical equalizers, noise gates, and compressors are simulated and automatically operated.

5.2.3 Setup

Before lectures are recorded, the user creates a so-called audio profile. It represents a fingerprint of the interplay of sound card, equipment and speaker. The profile is recorded using a multi-step wizard that guides the user through several steps, see Figure 5.2. This setup takes about three minutes and has to be done once per speaker and sound equipment. Each speaker uses their own audio profile for all recordings.

The setup screen asks to assemble the hardware as it is to be used in the lecture recording. The wizard detects the sound card and its mixing capabilities, see Figure 5.3. Using the operating system's mixer API, the sound card's input ports are scanned to find plugged-in recording devices. This is done by briefly reading from each port with gain set to maximum, while all other input lines are muted. The line with the highest noise level is assumed to be the input source. If noise level differences are below a certain threshold, the user is required to select the line manually. With a single source plugged in, this occurs only with digital input lines because they produce no background noise. At this stage, several hardware errors can also be detected, for example if noise is constantly at zero decibel there must be a short-circuit.

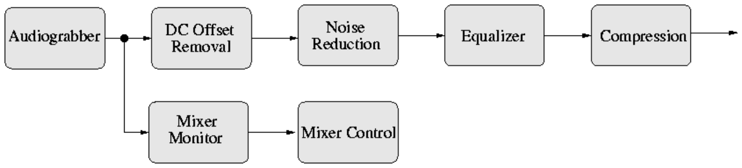


Figure 5.5: The chain of the audio signal during recording.

The audio system analyzer takes control over the sound card mixer. There is no need for the user to deal with the operating system's mixer.

The next step is to record the sound card background noise. The user is asked to remove any input device from the sound card.[3] A few seconds of noise are recorded. The signal is analyzed to detect possible hardware problems or handling errors. For example, overflows or critical noise levels result in descriptive warnings.

After recording sound card noise level, the user is asked to replug and switch on the sound equipment. Again, several seconds of "silence" are recorded and analyzed. Comparing this signal to the previous recording exposes several handling and hardware errors. For example, a recording device plugged into the wrong input line is easily detected.

After having recorded background noise, the user is asked to record phrases with special properties. The phrase choice is language-dependent. A phrase containing many plosives is used to determine the range of gain.[4] This measurement of the signal dynamics is used to adjust the automatic gain control. By adjusting sound card mixer input gain at the current port, the gain control levels out the signal. The average signal level should be maximized, but overflows must be avoided. If too many overflows are detected, or if the average signal is too low, the user is given a choice of possible improvements.

During the frequency test, a sentence containing several sibilants is recorded to figure out the upper-bound frequency. The system looks at the frequency spectrum to warn the user about equipment anomalies.

The final recording serves as the basis for a simulation. The user is asked to record a typical start of a lecture. The recording is filtered[5], compressed, and uncompressed again. Users can listen to their voice exactly as it will sound in the recording. If necessary, an equalizer allows experienced users to further fine-tune the frequency spectrum manually.[6] The time for filtering and compressing is measured. If this process takes too long, it is very likely that audio packets are lost during real recording due to a slow computer.

At the end of the simulation process, a report will be displayed, as shown in Figure 5.4. The report summarizes the most important measurements and grades sound card, equipment, and signal quality into the categories *excellent*, *good*, *sufficient*, *scant*, and *inapplicable*. The sound card is graded using back-

[3] On notebook computers this is not always possible, because built-in microphones cannot always be switched off. The wizard then adjusts its analysis process.

[4] In English, repeating the word "coffeepot" gives good results.

[5] The filters applied are described in Section 5.2.4.

[6] This might be assisted by supplying equalizer presets for different pitches of voice like bass and tenor, and by automatically preselecting the preset fits best.

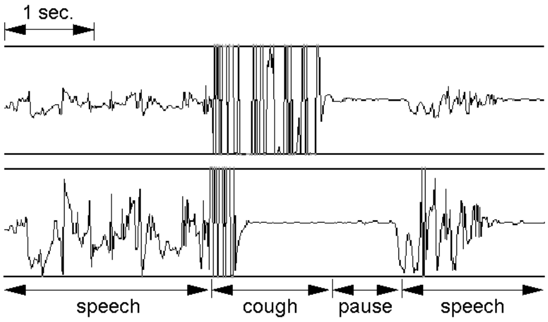


Figure 5.6: Without (above) and with (below) mixer control. The speech signal is expanded and the cough is leveled out.

ground noise and the card's DC offset[7] is calculated from the recordings. Grading the equipment is based on background noise recordings and frequency shape. This is only a rough grading, assisting non-expert users to judge the equipment and identify quality bottlenecks. Further testing would require the user to work with loop-back cables, frequency generators, and/or measurement instruments.

Among other information, the created profile contains all mixer settings, the equalizer settings, the recording, and the sound card's identification.

5.2.4 During Recording

For recording, the system relies on the equipment profile. If changes are detected, for example a different sound card, the system will display a warning at start up. Figure 5.5 illustrates the signal-processing chain.

The mixer settings saved in the profile are used to initialize the sound card mixer. The mixer monitor will display a warning if it detects a change in the hardware configuration, such as using a different input jack. It supervises the input gain in combination with the mixer control.

The mixer control uses the values of the dynamic test to level out the input gain using the sound cards mixer.[8] This makes it possible to level out voice-intensity variations. Coughs and sneezes, for example, are leveled out, see for example Figure 5.6. Note that the success of this method depends on the quality of the sound card's analog mixer channels. Sound cards with high-quality analog front panels, however, are becoming cheaper and more popular.

Mixer control reduces the risk of having feedback loops. Whenever a feedback loop starts to grow, the gain is lowered. As in analog compressors used in recording studios, the signal-to-noise ratio is lowered. For this reason, noise filters are required.

To reduce noise, the DC offset of the signal is removed, the sound card background noise level recorded in the profile is used as a threshold for a noise gate, and the equipment noise is taken as a noise fingerprint. The fingerprint phase is aligned with the recorded signal and subtracted in frequency space. This

[7] High DC offset implies low quality of the card's analog-to-digital converter.

[8] The analog pre-amplifiers of the mixer channels thus work like expander-compressor-limiter components used in recording studios.

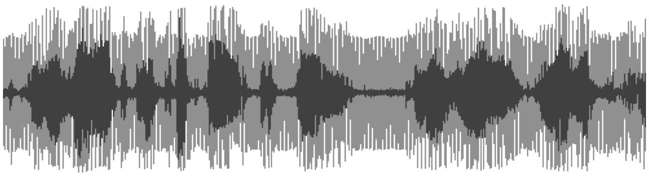


Figure 5.7: Three seconds of a speech signal with a 100 Hz sine-like humming before (light gray) and after filtering (dark gray).

removes any humming caused by electrical interference. Because the frequency and shape of the humming might change during a lecture[9], multiple noise fingerprints can be specified. The best match is subtracted [Bol79]. See Figure 5.7 for an example. It is not always possible to pre-record the humming, but if so, this method is superior to using electrical filters, which have to be fine tuned for a specific frequency range and often remove signal data more than is desirable.

Finally, Equalizer settings are applied before the normalized signal is processed by the codec.

As noted above, filtering also results in more efficient compression. Because noise and overflows are reduced, entropy also scales down and the compression can achieve better results.

In addition to controlling the signal for recording, the system checks the signal from the soundcard for the floor noise of the microphone. If the floor noise becomes significantly lower than expected according to the audio profile, it is likely that a problem occurred. Batteries of a wireless microphone may be running low, or the plug might been accidently pulled from the soundcard, or the lecturer might simply have forgotten to switch the microphone on. The system warns the lecturer by displaying a warning dialog, see Figure 5.8.

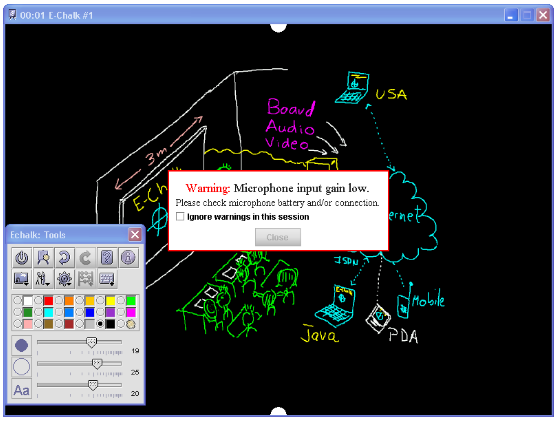
Of course, if the quality of the recording equipment (microphone and soundcard) is too low, even enhanced recording will not be enjoyable for the listener. However, first experiences [FKST04,FJK04b] showed that the system produced good quality when used with good amateur or semi-professional-grade equipment, which can be employed at significantly lower cost than professional audio equipment and audio technicians. In addition, the system's wizard already assisted a number of users in analyzing their audio hardware for shortcomings.

5.3 The WWV Video Server

The video subsystem called World Wide Video (WWV) was designed as an Internet streaming system that runs on any hardware or platform, just like WWR2/3. Explicitly included are small computers, such as handhelds or mobile phones. Since video data processing is more expensive than audio data processing, it was decided to create an asymmetric system. It was assumed that the server side has more computational performance at hand than the replaying client.

The bandwidth requirements for high-quality video are much higher than

[9] A typical situation that changes humming is when the light is turned on or off.



Warning: Microphone input gain low.
Please check microphone battery and/or connection.
☐ Ignore warnings in this session

Figure 5.8: The microphone floor noise level has sunk – batteries have to be changed.

for board or audio data, while the value as information channel is much lower.[10] Therefore, the E-Chalk system opted for a low-quality, low-bandwidth solution.

When the first version of WWV was developed, no method to capture video data in pure Java was available [FKR02]. Instead, native interfaces of the video hardware device were used, under Linux relying on Video4Linux 2 [101] and under Windows interfacing to Microsoft's DirectX 8.0a [62]. For efficiency reasons, the encoder was also realized as a native implementation. Recently, the WWV implementation became pure Java, utilizing the Java Media Framework (JMF) [44] to access video data.

Like the WWR3 audio server, the new video server has been developed within the SOPA component architecture described in Section 5.1.1. On startup of the server, the system creates a flow graph with video handling SOPA components as graph nodes from a graph description given as an XML file. Among other advantages, this keeps down development efforts to integrate new filters or support other output formats.[11]

For a description of the buffering strategy used in the WWV client, see Section 7.4.

Codec

The video stream is encoded by a very simple differential frame coder. The first frame recorded or sent when a client connects is an I-Frame (a full image). All subsequent frames are difference frames. The frames are divided into blocks of 8×8 pixels and only blocks that changed significantly are stored.

The change is measured as a sum of the Euclidean distances of all pixels

[10] From the very beginning of E-Chalk, the system focussed on transmitting the information by board content and voice stream. This philosophy is backed by experiences from other projects: the lecturer's voice stream is reported to be the most important information channel for learners [BGL96, EGE97], followed by the whiteboard/blackboard/slide stream [MO02], with the video stream of the lecturer mainly being of interest to create a sort of social context.

[11] The current implementation of the WWV server can already produce non-E-Chalk formats, for example QuickTime video.

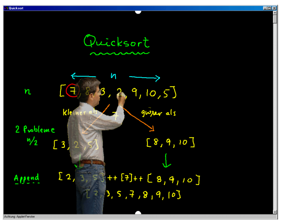


Figure 5.9: Semi-transparent instructor over board image (montage).

in the YUV color space, with the Y component given a higher weight. The relevant blocks are encoded in JPEG format.[12] Decoding is easily manageable in the Java Applet, because the JPEG decoder is available in standard libraries even for older JVMs.

In the teaching situation recorded for E-Chalk purposes, where only a small area of the picture has to be updated (mainly due to gestures and mimics of the lecturer), this very simple compression performs very well. The compression ratio obtained in experiments is roughly 40:1. In recordings with the E-Chalk system, a video resolution of 192×144 (quarter NTSC) and 4 frames per second was the standard setup to obtain a bandwidth of 64 kbps.

5.4 Video and Board Image Combined

E-Chalk lectures which include a video of the instructor share a psychological problem with conventional distance lectures consisting of slides plus video stream. Although every human being has only one locus of attention [Baa88, Ras00][13], the attention of the remote student is led to two areas of the screen: the video window and the board or slides area. Hence it was tried to separate the video image of the lecturer from the background. The image of the instructor can then be laid over the board, creating the impression that the lecturer is working directly on the screen of the remote student. Mimics and gestures of the instructor would appearing in direct relation to the board content. Ideally, the image of the lecturer can be made semi-transparent or even turned off, as illustrated in Figure 5.9.

The first approach to separate the lecturer's image from the background used an observation from the WWV video encoding: blocks stored in the difference frames almost always contained the lecturer's image. Changes in board content rarely result in differences the encoder rates as significant. However, the difference frames seldom contained the complete image of the instructor, and parts of the background might be briefly obscured by a moving lecturer only to appear again later. To compensate for these effects, the background was "learned" by storing repeated (similar) blocks into a block cache with a least-recently-used

[12] Strictly speaking, the term "JPEG format" is incorrect. The right term should be JFIF (JPEG file interchange format), whereas the specification can be found in [ISO92]. However, the term "JPEG format" is used more commonly.

[13] Some results from research on spatial attention suggest that it is actually possible to attend to two spatial locations simultaneously under certain circumstances [HK98, BCP99]. The model of assuming only a single locus may be simplistic, but concerning its application in user-interface design, it is preferred for practical purposes.

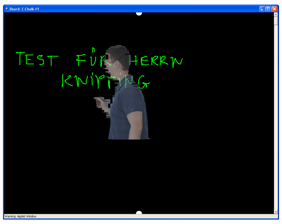


Figure 5.10: Screenshot of the experimental version for cutting out the lecturer. The video is already shown over an board image, but the lecturer's image is not yet correctly aligned with the board drawings.

strategy, keeping track of the number of recurrences. Those blocks with a high appearance count are likely to belong to the background, others are considered foreground blocks, i. e. the lecturer. For details, see [JFR04].

This algorithm is currently refined by combining the initial strategy with other strategies for classifying between background and lecturer, for example by looking at the color histograms of the blocks and by using edge-detection filters. Figure 5.10 shows an example result. Note that these approaches do not yet take any advantage of information about the board content in the background, which is available from the E-Chalk board or alternatively by taking screen snapshots.