Appendix F

E-Chalk's Video Format

This appendix describes the syntax of E-Chalk's video format. A conceptual explanantion can be found in Chapter 8. All integers defined herein are unsigned 32-bit big-endian, all shorts are unsigned 16-bit big-endian, and all bytes contain 8-bits unsigned data. The overall syntax of any E-Chalk video stream is:

```
<header>(<packetlength><packet>)*
```

The ⟨`header`⟩ is a 10-byte sequence and is described below. The ⟨`packetlength`⟩ is a short specifying the length in bytes of the following ⟨`packet`⟩. A ⟨`packet`⟩ is a gzipped [P. Deutsch, 1996] sequence of frames. A video stream can contain any number of packets. Archived video files may be accompanied by an *index file*, called `index.wwv`, residing at the same location as the video file. The index file associates a timestamp with a packet number and an offset inside the packet for faster random seek. The syntax is:

```
(<timestamp><packetno><offset>)*
```

A ⟨`timestamp`⟩ is an integer counting the milliseconds from the beginning of the recording. The ⟨`packetno`⟩ is an integer counting the packets from the beginning of the file, and ⟨`offset`⟩ specifies the position of the frame in the uncompressed packet (also a 4-byte big-endian integer). Video files greater than 4 GB cannot have an index file.

F.1 Header

The header is stored at the beginning of each file. If a lecture is appended to an older E-Chalk lecture, the header is substituted at the beginning of the old file, i. e., it is guaranteed that there are no headers in the middle of a video file. During a live transmission, the header is sent to any client that connects to the server. The header is thus the first sequence of bytes that every player receives. The syntax is explained in the following table.

Offset	Size (bytes)	Content	Description
0	2	"FU"	magic bytes
2	1	byte	'0'=window mode, '1'=board overlay
3	2	short	initial x-resolution in pixels
5	2	short	initial y-resolution in pixels
7	1	byte	initial framerate (frames per second)
8	1	byte	reserved for future use
9	1	byte	reserved for future use

In board overlay mode, the resolution information is ignored and the video is scaled to fit the board resolution.

F.2 Packet

The syntax of an uncompressed packet is defined as follows:

```
<packet>::=(<size><type><frame>)*
<type>::='0'|'1'|'2'|'3'
<frame>::=<i-frame>|<t-frame>|<0-frame>
```

The entry ⟨`size`⟩ is a three-byte descriptor specifying the size of the compressed image (high byte first, low byte last). The ⟨`type`⟩ is one byte describing the frame type: '0' stands for an I-Frame, '1' is obsolete and not supported any more, '2' stands for a T-frame, and '3' for a 0-frame. Higher numbers are reserved for future use and may be ignored at this time. The three frame types are described in the following.

I-Frames

I-Frames are optional. I-Frames are able to change the frame rate and the player resolution. This enables merging several video streams that were recorded with a different resolution and/or frame rate. In board overlay mode, resolution information is ignored and the video is scaled to fit the board resolution. Additionally, the color black is defined as transparent. I-Frames are encoded as follows:

Offset	Size (bytes)	Content	Description
0	2	short	new x-resolution in pixels
2	2	short	new y-resolution in pixels
4	1	byte	new framerate (frames per second)
5	⟨`size`⟩	bytes	JFIF encoded image data

T-Frames

T-Frames contain a transparency table that has one bit associated with each 8×8-pixel block in the image. The index in the table is canonically organized. It starts at the upper left corner and ends in the bottom right corner. If the bit is set, the corresponding block is to be drawn; if the bit is not set, the block is transparent. The size of the descriptor ds in bytes is $\lceil \frac{x \cdot y}{512} \rceil$ with x and y being the x and y resolution of the image, respectively. In board overlay mode, the color black is defined as transparent. T-Frames are encoded as follows:

Offset	**Size (bytes)**	**Content**	**Description**
0	ds	bytes	block transparency descriptor
ds	$\langle$`size`$\rangle$	bytes	JFIF encoded image data

0-Frames

0-Frames are only defined by their type. They cause no drawing for a frame rate's part of a second.