

# Howto der MJPEGTOOLS

Autoren: siehe Kontakt und Danksagung

21. November 2004

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>4</b>
1.1	Bücher . . . . .	6
<b>2</b>	<b>Begriffsklärungen</b>	<b>7</b>
<b>3</b>	<b>Bezugsquellen</b>	<b>10</b>
<b>4</b>	<b>Wiedergabe der erzeugten Videos</b>	<b>12</b>
<b>5</b>	<b>Aufnahmen mit lavrec</b>	<b>13</b>
5.1	Generelles . . . . .	13
5.2	Bedienung . . . . .	13
5.2.1	Programmaufruf . . . . .	13
5.2.2	Ausgabe während der Aufnahme . . . . .	13
5.2.3	Beispiele . . . . .	14
5.2.3.1	Beispiel 1 . . . . .	14
5.2.3.2	Beispiel 2 . . . . .	14
5.2.3.3	Beispiel 3 . . . . .	15
5.2.3.4	Beispiel 4 . . . . .	15
5.2.3.5	Beispiel 4 . . . . .	16
5.3	Fehler, Fehlererkennung und Fehlerbehebung . . . . .	16
5.3.1	Die wichtigsten Fehler . . . . .	16
5.3.2	Fehlererkennung . . . . .	17
5.3.3	Fehlerbehebung . . . . .	18
<b>6</b>	<b>Videofilme von einzelnen Bildern erzeugen</b>	<b>20</b>
<b>7</b>	<b>Abspielen des aufgenommenen Films</b>	<b>22</b>
<b>8</b>	<b>Bearbeitung des Videos</b>	<b>24</b>
8.1	Bearbeiten mit glav . . . . .	24
8.2	Mehrere Videos zu einem vereinen . . . . .	25
8.3	Ton auskoppeln . . . . .	26
8.4	Einzelbilder aus dem Video auskoppeln . . . . .	26
8.5	Überblendungen erzeugen . . . . .	27

<b>9</b>	<b>Einen Film in MPEG- oder DIVx Videos umwandeln</b>	<b>30</b>
9.1	Allgemeines . . . . .	30
9.2	Grundsätzliches zur Tonkomprimierung . . . . .	31
9.3	Grundsätzliches zur Videokomprimierung . . . . .	32
9.3.1	Skalieren . . . . .	33
9.4	Grundsätzliches zum Zusammenführen von komprimiertem Ton und Bild . .	34
9.5	Videofilme im mpeg1-Format erzeugen . . . . .	35
9.5.1	Beispiele für eine Audiodatei im mpeg1-Format . . . . .	35
9.5.2	Beispiele für eine Videodatei im mpeg1-Format . . . . .	36
9.5.3	Beispiele für das Zusammenführen der Dateien (multiplexen) im mpeg1- Format . . . . .	37
9.6	Videofilme im mpeg2-Format erzeugen . . . . .	38
9.6.1	Beispiel für eine Audiodatei im mpeg2-Format . . . . .	38
9.6.2	Beispiele für eine Videodatei im mpeg2-Format . . . . .	38
9.6.2.1	Werte der Option <b>-q bei Filmen mit variabler Bitrate (VBR)</b> . . . . .	39
9.6.2.2	Kompression für das Wiedergabeziel Fernsehen (interlaced) oder Monitor (progressive) . . . . .	40
9.6.3	Beispiel für das Zusammenführen der Dateien (multiplexen) im mpeg2- Format . . . . .	41
9.7	Video-CD (VCD) erzeugen . . . . .	41
9.7.1	Beispiel für die Tonkomprimierung für eine VCD . . . . .	41
9.7.2	Beispiel für die Videokomprimierung für eine VCD . . . . .	42
9.7.3	Beispiel für das Zusammenführen von Ton und Bild für eine VCD . .	42
9.7.4	Erzeugung der VCD . . . . .	43
9.7.5	MPEG-Filme speichern . . . . .	43
9.8	Super Video CD (SVCD) erzeugen . . . . .	44
9.8.1	Beispiel für die Tonkomprimierung für eine SVCD . . . . .	44
9.8.2	Beispiel für die Videokomprimierung für eine SVCD . . . . .	44
9.8.3	Beispiel für das Zusammenführen von Ton und Bild für eine SVCD .	46
9.8.4	Erzeugung der SVCD . . . . .	46
9.9	Erzeugung einer DVD . . . . .	46
9.9.1	Beispiel für die Tonkomprimierung für eine DVD . . . . .	47
9.9.2	Beispiel für die Videokomprimierung für eine DVD . . . . .	47
9.9.3	Beispiel für das Zusammenführen von Ton und Bild für eine DVD . .	47
9.9.4	Erzeugung der DVD . . . . .	48
9.10	DIVX-Videos erzeugen . . . . .	48
<b>10</b>	<b>Filmoptimierung</b>	<b>50</b>
10.1	Einsatz von Filterprogrammen . . . . .	50
10.2	Skalierung und Offset-Korrektur . . . . .	53
10.3	Konversion der Bildfolgefrequenz . . . . .	54
<b>11</b>	<b>Dekomprimierung von mpg-Filmen</b>	<b>56</b>
11.1	Dekomprimierung mit MPlayer . . . . .	56
11.2	Dekomprimierung mit mpeg2dec . . . . .	57
11.3	Weitere Formatwandlungsmöglichkeiten von mpeg2-Filmen . . . . .	57
<b>12</b>	<b>Abwägung Qualität gegen Geschwindigkeit</b>	<b>62</b>
12.1	Filme für Software-player und Disk Playback erzeugen . . . . .	62

<b>13 SMP und verteiltes Komprimieren</b>	<b>63</b>
<b>14 Kontakt und Danksagung</b>	<b>65</b>

# 1 Einführung

Die **mjpegtools** sind ein Programmpaket, das es erlaubt, bei unterstützter Video (v4l und v4l2) Karte Filme aufzunehmen, abzuspielen, zu bearbeiten, zu komprimieren und in andere Formate zu konvertieren. Bei den **mjpegtools** handelt es sich durchwegs um Kommandozeilen orientierte Programme (Ausnahme: **glav**). Es sind im einzelnen folgende Programme:

- Programm **lavrec**  
Mit diesem Programm können Videos aufgenommen werden.
- Programme **lavplay** und **glav**  
Mit diesen Programmen können die Aufnahmen wiedergegeben und bearbeitet werden werden.
- Programm **lav2wav**  
Dekomprimiert den Ton von einem File nach stdout.
- Programm **lav2yuv**  
Dekomprimiert die Bilder von einem File nach stdout. Die Bilder sind im yuv4:2:0 Format.
- Programm **yuvdenoise**  
Beseitigt Bildrauschen durch filtern.
- Programm **yuvscaler**  
Skaliert das Bild auf die gewünschte Größe und kann die aktive Größe festlegen.
- Programm **lav2mpeg**  
Ein Script, das die einzelnen Kommandos verpackt. Damit kann das Komprimieren mit nur einen Aufruf erledigt werden.
- Programm **mp2enc**  
Komprimiert den Ton in das mp2-Format.
- Programm **mpeg2enc**  
Komprimiert die Bilder vom yuv4:2:0 in das mpeg1/2 Format.
- Programm **mplex**  
Zusammenführung des komprimierten Ton und Videos zu einem MPEG File, der dann mit MPEG-playern (xine, mplayer, vlc u.a.) abgespielt werden kann. Eine genaue Beschreibung der Programme und deren Optionen findet man in der Manpage des jeweiligen Programmes.

Dies sind die zunächst wichtigsten Programme zum Erzeugen eines mpg-Films. Darüberhinaus enthalten die mjpegtools eine Reihe von weiteren Programmen, auf die an anderer Stelle eingegangen wird.

Soll der erzeugte mpg-Film auf einem DVD-Spielgerät am Fernseher abgespielt werden, dürfen nicht alle Optionen, die MPEG verwendet, benutzt werden. Deshalb gibt es vordefinierte Profile, wo viele Optionen schon richtig gesetzt sind, damit die DVD-Spieler das Video lesen können. Es ist möglich, eine VCD, SVCD oder eine DVD daraus zu erzeugen. Die Erzeugung einer VCD oder SVCD kann einfach mit dem Programm **VCDImager** erfolgen. Bei Erzeugung einer DVD wird man zu dem Programm **dvdauthor** greifen. Dazu weiter unten mehr.

Der typische Ablauf wenn man die Videos von einer Quelle aufnimmt. Und das Video dann bearbeitet sieht etwa wie im Bild angegeben aus. Im Bild sieht man auch die Schnittstellen von und zu anderen Programmpaketen diese sind in grau gehalten. Alle anderen Teile kann das mjpegtools Paket abdecken.

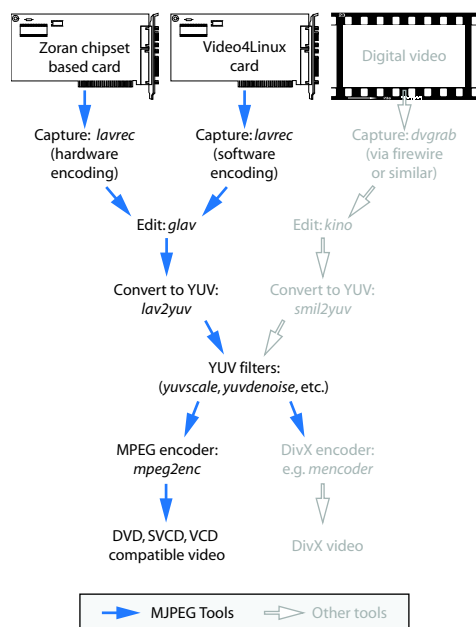


Abbildung 1.1: Video Codierung Flussdiagramm

Vor Beginn der Nutzung der **mjpegtools** müssen die Environment Variablen gesetzt werden, sofern die Einstellungen nicht den Voreinstellungen entsprechen, oder wenn man mehrere tv-Karten hat, die man gleichzeitig nutzen will. Es handelt sich um folgende Environment Variablen:

**LAV\_VIDEO\_DEV**

Voreinstellung ist /dev/video

**LAV\_AUDIO\_DEV**

Voreinstellung ist /dev/dsp

**LAV\_MIXER\_DEV**

Voreinstellung ist /dev/mixer

Gesetzt werden die Variablen mit dem Befehl z. B.

**export LAV\_VIDEO\_DEV=/dev/video**

Wie man Enviroment Variablen setzt, ist in der Beschreibung der Shell genauer beschrieben.

## 1.1 Bücher

Hier noch eine Auswahl an Büchern die man als weitere Informationsquellen nutzen kann. Zuerst einige englische Bücher:

- Digital Video and HDTV von Charles Poyton (ISBN 1-55860-792-7)
- Digital Video Compression von Peter Symes (ISBN 0-07-142487-3)
- Video Demystified von Keith Jack (ISBN 1-878707-56-6)

Und nun in Deutsch:

- Fernsehtechnik von Rudolf Mäusl (ISBN 3-7785-2374-0)
- Professionelle Videotechnik - analoge und digitale Grundlagen von U. Schmidt (ISBN 3-540-43974-9)
- Digitale Film- und Videotechnik U. Schmidt (ISBN 3-446-21827-0)

## 2 Begriffsklärungen

- **Interlacing**

Das menschliche Auge ist in der Lage, etwa 16 Einzelbilder pro Sekunde zu unterscheiden. Höhere Bildfolgen werden vom Auge nicht mehr aufgelöst und als bewegtes Bild wahrgenommen. Bei der Film- und Videotechnik benutzt man deshalb Bildwechselfrequenzen, die deutlich höher sind als 16 Bilder pro Sekunde. Im einzelnen sind dies

- beim Film:  $24 \frac{\text{Bilder}}{\text{s}}$ ,
- bei der PAL-Norm (gültig in vielen europäischen Ländern):  $25 \frac{\text{Bilder}}{\text{s}}$ ,
- bei der NTSC-Norm (US amerikanische Norm, auch gültig in Kanada, Japan und anderen Ländern): 30000 : 1001 oder 24000 : 1001  $\frac{\text{Bilder}}{\text{s}}$ ,
- bei der SECAM-Norm: (gültig in Frankreich und weiteren europäischen Ländern):  $25 \frac{\text{Bilder}}{\text{s}}$ .

Da das Auge auch bei einer Bildwechselfrequenz von  $25 \text{ Bilder/s}$  den Bildwechsel vor allem der hellen Bildteile noch als „Flimmern“ wahrnimmt, bedient man sich beim Fernsehen eines Tricks, um die Bildwechselfrequenz scheinbar zu erhöhen und so ein flimmerfreies Bild zu bekommen. Das Fernsehbild wird zeilenweise von links oben nach rechts unten geschrieben. Statt jeweils ein Vollbild zu senden, überträgt man zwei Halbbilder, und zwar zunächst die Zeilen 1, 3, 5, ... als erstes Halbbild und anschließend die Zeilen 2, 4, 6, ... als zweites Halbbild. Dadurch wird dem Auge eine Bildwechselfrequenz von  $50 \text{ Bilder/s}$  (PAL, SECAM) bzw. ca  $60 \text{ Bilder/s}$  (NTSC) vorgetäuscht und das Bild erscheint flimmerfrei. Dies wird in der Fachsprache als **Interlacing** bezeichnet. Näheres zum Interlacing siehe die einschlägige Literatur z. B. Mäusl, Fernsehtechnik oder U. Schmidt, Videotechnik

- **Bild- und Tongrößen und zu erwartende Bitraten**

Gleichungen zur Errechnung der Bitrate:

- **Audio für CD-Qualität:**

$$\text{Bitrate} = \frac{44100 \text{ samples/s} \cdot 2 \text{ Kanäle} \cdot 16 \text{ Bit Breite}}{8 \frac{\text{Bit}}{\text{Byte}} \cdot 1024 \text{ B/kB}} = 172,7 \text{ kB/s}$$

- **Video:**

$$\text{Bitrate} = \frac{\text{Breite} \cdot \text{Höhe} \cdot \text{Bildfolgefrequenz} \cdot \text{Qualität}}{\text{Kartentype} \cdot 1024 \text{ B/kB}}$$

Für die Kartentype ist bei der Buz 200 einzusetzen, bei allen andern Karten wie DC10/DC30/LML33 ist der Wert 100 zu verwenden.



\* **PAL halbe Größe:**

$$\text{Bitrate} = \frac{352 \cdot 288 \cdot 25 \cdot 80}{200 \cdot 1024} = 990 \text{ kB/s}$$

\* **PAL volle Größe:**

$$\text{Bitrate} = \frac{720 \cdot 576 \cdot 25 \cdot 80}{200 \cdot 1024} = 4050 \text{ kB/s}$$

\* **NTSC halbe Größe:**

$$\text{Bitrate} = \frac{352 \cdot 240 \cdot 30 \cdot 80}{100 \cdot 1024} = 1980 \text{ kB/s}$$

\* **NTSC volle Größe:**

$$\text{Bitrate} = \frac{720 \cdot 480 \cdot 30 \cdot 80}{100 \cdot 1024} = 8100 \text{ kB/s}$$

- **MPEG1 und MPEG2**

International genormte Verfahren zur Datenreduktion. MPEG ist die Abkürzung von *Moving Pictures Expert Group*. Um die sehr große Datenrate bei der Bildwiedergabe zu reduzieren, werden die Bilder durch eine sogenannte *Hybride Diskrete Kosinus-Transformation* im Datenfluß reduziert. Das dabei auftretende Problem sind Bilder von sich (schnell) bewegenden Objekten. Um auch bei den reduzierten Bildern die Bewegungen ohne „Flackern“ wiedergeben zu können, muß der Gang der zukünftigen Bewegung vorausgeahnt werden. Dazu wird während der Kompression der Bilder immer einige Bilder in die Zukunft geschaut, um zu wissen, wohin sich die Bewegung entwickeln wird. Der MPEG 2-Standard ist eine Weiterentwicklung des MPEG 1-Standards.

- **Platzhalter im Dateinamen**

Häufig ist es erwünscht, die erzeugten Dateien eine gewünschte Größe nicht überschreiten zu lassen. Will man z. B. den erzeugten Film auf eine CD mit 700 MB Platz brennen, sollte die Datei tunlichst kleiner als 700 MB sein. Um dies zu gewährleisten, ermöglichen die **mpegtools** durch einen Platzhalter im Dateinamen bei Überschreitung einer vorbestimmten Dateigröße automatisch eine zweite (dritte) Datei anzulegen. Vergibt man z.B. beim Aufnehmen mit **lavrec** den Dateinamen Datei.avi für die aufzunehmende Datei, stoppt **lavrec** die Aufnahme automatisch bei Überschreiten einer Dateigröße von z. B. 1,6 GB (Die Größe ist format- und architekturabhängig). Soll das nicht erfolgen, muß im Dateinamen der Platzhalter **%02d** eingefügt werden, im Beispiel sollte also der Dateiname **Datei%02d.avi** lauten. Dann werden automatisch Dateien mit den Namen Datei01.avi, Datei02.avi, usw. angelegt, bis die Aufnahme gestoppt wird. Dies gilt sinngemäß auch für die anderen Programme der **mjpegtools**.

- **Fileformate AVI und Quicktime**

Beide Formate sind sogenannte Containerformate, die beschreiben, wie die Audio und Video Streams, die in einer Datei gespeichert sind, beschaffen sein müssen. Dabei kann man für Audio und Video verschiedene Codecs verwenden. Die **mjpegtools** sind in der Lage, AVI- und Quicktime-Formate zu verarbeiten, die als Video Codec mjpeg- (Motion JPEG) oder DV Type 1 und als Audio das WAV-Format verwenden. Wenn man andere Codecs in dem AVI- oder Quicktime-Format hat, sind die nicht von den **mjpegtools** lesbar. AVI ist durch die interne Struktur auf 32Bit Architekturen (typischer PC) auf 2 GB Filegröße beschränkt, bei Quicktime gibt es diese Limitierung nicht. Für AVI gibt es mehrere verschiedene Erweiterungen, die das 2 GB Limit aufheben. Diese wurden aber noch nicht implementiert, da sie grundsätzliche Probleme nicht beheben, wie z. B. das Bearbeiten von einem File, während noch aufgenommen wird.

- **Bandbreite**

Dies ist ein anderes Wort für eine Datenmenge in einer bestimmten Zeiteinheit. Die Einheit ist meist *kByteproSekunde* ( $\frac{kB}{s}$ ). Wenn man in einer Sekunde 1 *MByte* (1 *MB*) übertragen kann, oder bei einem Video in einer Sekunde 1 *MByte* an Daten anfällt, spricht man von einer benötigten Bandbreite von 1 *MByte*.

- **GOP**

Die Englische Langform heißt Group of Pictures, was etwa Gruppe von Bildern in Deutsch entspricht. MPEG faßt immer eine bestimmte Anzahl von Frames zu einer Gruppe zusammen, wobei das erste Bild der Gruppe immer mit einem I-Frame beginnt, der den gesamten Frame speichert und somit am meisten Platz verbraucht. Dann gibt es noch P-Frames, die zwischen I-Frames sind und die Änderungen zum vorherigen I oder P-Frame speichern. Die B Frames sind nur bei VCD verpflichtend vorgeschrieben und enthalten die Änderung zum letzten und zum nächsten I oder P Frame. Bei MPEG ist es je nach Verwendung teilweise dem Encoder berlassen, wie der die I/P/B Frames verwendet um möglichst viel Platz sparen.

- **Gamma** Gamma, eigentlich sollte bei PAL ein Gamma von 2.8 verwendet werden. Das wird aber oft als zu hoch angesehen, und deshalb wird bei PAL auch der NTSC Gamma Wert von 2.2 verwendet. Die mjpegtools verändern das Gamma nicht und belassen es genauso wie es ist (shit in shit out Prinzip). Wenn man das Gamma ändern will sollte man dies machen bevor der Film von den mjpegtools verarbeitet wird. Ist das nicht mehr machbar so kann man mit yuvcorrect, das Gamma korrigieren. Dies ist im allgemeinen aber nicht notwendig

## 3 Bezugsquellen

Die mjpegtools können unter

**<http://mjpeg.sourceforge.net>**

heruntergeladen werden.

Zum Erzeugen von VCD und SVCD kann der VCDImager unter

**<http://www.vcdimager.org>**.

heruntergeladen werden.

Ein Programm zum Erzeugen von DVD gibt es unter

**<http://dvdauthor.sourceforge.net>**.

DVIX Videos kann man mit dem Programm **mencoder** erzeugen, zu finden unter

**<http://www.mplayer.hu>**.

MAC-Besitzer, die die **mjpeg tools** verwenden möchten, finden unter folgenden Adressen Informationen:

**[http://www.sjoki.uta.fi/~shmhav/SVCD\\_on\\_a\\_Macintosh.txt](http://www.sjoki.uta.fi/~shmhav/SVCD_on_a_Macintosh.txt),**

**<http://homepage.mac.com/rnc/>**

Wenn man die mjpeg tools auf einem MAC compilieren möchte, oder sich nur die compilieren Versionen von mpeg2enc und mplex besorgen will, so findet man die hier:

<http://mjpeg.sf.net/MacOS/>

Bei

**<http://www.dvdrhelp.com>**

kann nachgesehen werden, was der DVD-Player abspielen kann.

Ein weitere Dokumentation über das Aufnehmen codieren und brennen ist hier zu finden bei dem Titel Video from an Analogue Camcorder:

<http://mandrake.vmlinux.ca/bin/view/Main/VideoApps>

Eine GUI für die mjpegtools namens Linux Video Studio findet man hier:

<http://ronald.bitfreak.net>

## 4 Wiedergabe der erzeugten Videos

Die mit den **mpegtools** erzeugten Videos sind beispielsweise mit folgenden Programmen abspielbar:

<b>dxr2</b>	<b>hardware decoder card</b>
<b>xine</b>	<a href="http://xine.sourceforge.net">http://xine.sourceforge.net</a>
<b>oms</b>	<a href="http://www.linuxvideo.org">http://www.linuxvideo.org</a>
<b>dvdview</b>	<a href="http://rachmaninoff.informatik.uni-mannheim.de/dvdview">http://rachmaninoff.informatik.uni-mannheim.de/dvdview</a>
<b>xmovie</b>	<a href="http://heroinewarrior.com/xmovie.php3">http://heroinewarrior.com/xmovie.php3</a>
<b>mplayer</b>	<a href="http://mplayer.sourceforge.net">http://mplayer.sourceforge.net</a>
<b>vlc</b>	<a href="http://www.videolan.org">http://www.videolan.org</a>
<b>nur MPEG1: mtv</b>	<a href="http://www.mpeg1.com">http://www.mpeg1.com</a>
<b>nur MPEG1: gtv</b>	<a href="http://packages.debian.org/stable/graphics/smpeg-gtv.html">http://packages.debian.org/stable/graphics/smpeg-gtv.html</a>

### MS Media Player

### Diverse SW DVD Player

Der MS Media Player scheint MPEG 1 lieber abzuspielen, wenn sie mit -f 1 gemultiplexed wurden.

MJPEG-AVI Videos, die mit dem streamer tool (bttv basierte Karte) aus dem xawtv Paket aufgenommen wurden, können zwar mit den **mjpegtools** verarbeitet werden, jedoch ist ein Hardware Playback auf Zoran basierten Karten aufgrund von Beschränkungen der ZORAN Hardware nicht möglich.

Mit Nuppel Video (<http://frost.htu.tuwien.ac.at/~roman/nuppelvideo/>) aufgenommene Videos können auch mit den **mjpegtools** verarbeitet werden.

Die Rohaufnahmen (avi-files) können neben den zu den **mjpegtools** gehörenden Programmen **lavplay** und **glav** auch mit den Programmen **xanim**, **mplayer**, oder **vlc** wiedergegeben werden.

# 5 Aufnahmen mit lavrec

## 5.1 Generelles

**lavrec** ist ein Kommandozeilen orientiertes Programm für das Anfertigen von Aufnahmen von laufenden Fernsehprogrammen, das für **ZORAN**-basierte tv-Karten geschrieben wurde. Durch die Option (s.u.) `–software-encoding` ist es aber möglich, auch mit **bttv**-basierten Karten mit **lavrec** aufzunehmen. Während der Aufnahme mit **lavrec** darf kein anderes Programm wie z. B. **xawtv** oder **kwintv** auf die bttv basierte Karte zugreifen. Das hat zur Folge, daß man während der eigentlichen Aufnahme am Rechner das Bild nicht verfolgen kann. Die auf dem Zoran Chipset basierten Karten haben diese Einschränkung nicht. Die Aufnahme kann jederzeit mit dem Befehl **Strg-C** (früher hat man **Ctrl-C** geschrieben) gestoppt werden.

## 5.2 Bedienung

### 5.2.1 Programmaufruf

Der Programmstart erfolgt aus der Kommandozeile mit dem Befehl

```
> lavrec [Optionen] Dateiname(n)
```

### 5.2.2 Ausgabe während der Aufnahme

Nach dem Starten einer Aufnahme erscheint in der Kommandozeile eine Ausgabe, die etwa wie folgt aussieht:

```
0.06.14:22 int:00040 lst:0 ins:0 del:0 ae:0 td1:=0.014 td2=0.029
```

Die einzelnen Größen bedeuten:

- **0.06.14:22**: Abgelaufene Zeit seit Start der Aufnahme
- **int**: Zeitintervall zwischen zwei Bildern. Sollte bei 33 (NTSC) oder 40 (PAL/SECAM) liegen. Wenn der Wert ein anderer ist, hat man eine schlechte Aufnahme und/oder viele verlorene Bilder.
- **lst**: Steht für lost frames (verlorene Bilder). Das ist die Anzahl der Bilder, die beim Aufnahmen nicht gespeichert werden konnten. Das kann an einer zu langsamen Festplatte oder zu hoher CPU Belastung liegen. Bei vielen verlorenen Bildern sollte man mit kleinerer Auflösung und niedrigerer Qualität aufnehmen.

- **ins, del:** Steht für inserted or deleted frames (eingefügte oder gelöschte Bilder). Dies ist normal, wenn sie nicht in großer Anzahl auftreten. Worauf viele eingefügte oder gelöschte Bilder zurückzuführen sind, kann nicht allgemeingültig beantwortet werden. Es kann helfen, wenn man Optionen angibt, die das System weniger belasten. Auch die Verwendung einer anderen Soundkarte kann helfen.
- **ae:** audio errors, sollte immer 0 sein.
- **td1, td2:** Steht für time difference. Gibt die Zeitdifferenz zwischen Ton und Bild an. Sollte immer bei 0 liegen, es sei denn, man hat die Synchronisationkorrektur ausgeschaltet (Option `-c 0`).

### 5.2.3 Beispiele

Es werden im folgenden einige exemplarische Beispiele für Aufnahmen mit `lavrec` wiedergegeben.

#### 5.2.3.1 Beispiel 1

```
> lavrec -f a -i P -d 2 Datei.avi
```

`lavrec` beginnt sofort mit der Aufnahme von einer Zoran-Chip basierten Karte via hardware. Die einzelnen Optionen bedeuten:

**-f a:** `lavrec` erzeugt eine AVI-Datei als Ausgangsformat.

**-i P:** aufgenommen wird über den SVHS-Eingang mit dem Eingangsformat PAL.

**-d 2:** die Bilder werden mit halber PAL-Auflösung (352x288, tatsächliche Auflösung Karten und Videonorm abhängig) aufgenommen.

**Datei.avi:** Name des erzeugten Files. Die Aufnahme wird automatisch gestoppt, wenn die Datei eine Größe von ca. 1,6 GB erreicht hat.

Die Aufnahme wird mit Strg-C beendet. Manchmal kann es notwendig sein, statt der Option `-f a` die Option `-f A` zu verwenden, z. B. wenn man Interlacing-Probleme hat.

#### 5.2.3.2 Beispiel 2

```
> lavrec -f q -i n -d 1 -q 80 -s -l 80 -R 1 -U Datei.avi
```

`lavrec` beginnt sofort mit der Aufnahme von einer Zoran-Chip basierten Karte via hardware. Die einzelnen Optionen bedeuten:

**-f q:** `lavrec` erzeugt eine Quicktime-Datei als Ausgangsformat.

**-i n:** es wird über den composite-in Eingang im NTSC-Format aufgenommen.

**-d 1:** es wird ein Video mit voller NTSC-Auflösung (720x480) erzeugt.

**-q 80:** die Qualität der einzelnen Bilder wird auf 80 % gesetzt (Voreinstellung: 50 %).

**-s:** der Ton wird in Stereo aufgenommen.

**-l 80:** der Mischer der Soundkarte wird auf 80 % gesetzt.

**-R 1:** für die Tonaufnahme wird der Line-In-Eingang der Soundkarte verwendet.

**-U:** Lavrec verwendet read statt mmap Mode bei der Tonaufnahme. Das wird benötigt, wenn die Soundkarte mmap-Aufnahmen nicht unterstützt.

**Datei.avi:** Name des erzeugten files. Die Aufnahme wird automatisch gestoppt, wenn die Datei eine Größe von 1,6 GB erreicht hat, wenn man AVI als Fileformat wählt. Bei Quicktime gibt es dieses Limit nicht. Die Aufnahme wird mit strg-c beendet.

Den Mischer setzen funktioniert nicht bei jeder Soundkarte. Werden zwei Aufnahmen mit unterschiedlichen Lautstärkeinstellungen gemacht und beide Aufnahmen sind bei der Wiedergabe gleich laut, funktioniert die Mischereinstellung nicht. Dann muß man den Mischer mit einem Mixer-Programm einstellen. Danach sollte man sollte mit lavrec nur noch mit der Option -l -1 aufnehmen.

Welche Größe das aufgenommene Bild hat, hängt von der verwendeten Karte ab. Bei voller Auflösung (-d 1) bekommt man diese Größen für PAL: BUZ und LML33: 720x576 und bei DC10 und DC30: 768x576

### 5.2.3.3 Beispiel 3

```
> lavrec -w -f a -i S -d 2 -l -1 Datei%02d.avi
```

Aufnahme von ZORAN-Chip basierten Karten via hardware.

**-w:** lavrec wartet mit dem Beginn der Aufnahme, bis die Enter-Taste gedrückt wird.

**-f a:** es wird eine AVI-Datei als Ausgabeformat erzeugt.

**-i S:** es wird über den SVHS-Eingang im SECAM-Format aufgenommen. (SECAM Aufnahme über den composite-Eingang: -i s).

**-d 2:** die Aufnahme erfolgt mit halber Auflösung.

**-l -1:** die Mischereinstellungen werden nicht verändert.

**Datei%02d.avi:** lavrec legt die Dateinamen selber fest, beginnend mit Datei01.avi. Sobald die Datei 1,6 GB groß ist (nach etwa 20 min Aufnahme), wird eine neue Sequenz mit dem Namen Datei02.avi erzeugt. Das wird solange fortgesetzt bis die Nummer 99 erreicht ist, oder bis die Aufnahme gestoppt wird, oder die Festplatte voll ist.

### 5.2.3.4 Beispiel 4

```
> lavrec -f a -i t -q 80 -d 2 -t 3600 -C europe-west:SE20 Datei%02d.avi
```

Lavrec beginnt sofort mit der Aufnahme von einer Zoran-Chip (nur bei den Matrox Marvel) basierten Karte via Hardware. Die einzelnen Optionen bedeuten:



**-f a:** es wird eine AVI-Datei als Ausgabeformat erzeugt.

**-i t:** es wird der Tuner der Marvelkarte als Eingang benutzt.

**-q 80:** die Qualität der einzelnen Bilder wird auf 80 % gesetzt (Voreinstellung: 50 %).

**-d 2:** die Aufnahme erfolgt mit halber Auflösung.

**-t 3600:** lavrec nimmt für die Dauer von 1 h (3 600 s) auf und beendet dann die Aufnahme selbsttätig.

**-C europe-west:SE20:** es wird vom Kanal SE20 aufgenommen. Setzt den TV-Kanal des Tuner-Eingangs (-i t und -i T). Das funktioniert derzeit nur bei den Matrox Marvel G200/G400-Karten und der Matrox Millenium G200/G400 mit dem Rainbow Runner Aufsatz. Am vollen Support der bttv-Karten wird noch gearbeitet. Mehr Infos, wie man den Tuner zum Laufen bringt, findet sich bei <http://marvel.sourceforge.net>.

#### 5.2.3.5 Beispiel 4

```
> lavrec -f a -i p -g 352x288 -q 80 -l 70 -R 1 --software-encoding Datei.avi
```

Die beiden neuen Optionen sind

**-g 352x288:** bestimmt die Bildgröße des aufgenommenen Videos.

**–software-encoding:** die Aufnahme erfolgt via software-encoding der Bilder, die von der Videokarte kommen. Mit dieser Option ist es möglich, von einer bttv basierten Karte Videos aufzunehmen. Die Prozessorlast ist dabei sehr hoch. Diese Option funktioniert nur für generische video4linux Karten (wie z. B. die booktree-848/878 oder Conexant basierten Karten). Bei Zoran basierten Karten funktioniert diese Option nicht.

## 5.3 Fehler, Fehlererkennung und Fehlerbehebung

Fehler treten meist durch das Interlacing auf. Die mjpegtools sind darauf ausgelegt, die Videodaten bildweise abzuarbeiten. Da interlaced Bilder aber jeweils aus zwei Halbbildern bestehen, dem top field mit der 0., 2., 4., ... Zeile und dem bottom field mit der 1., 3., 5., .... Zeile, ist es wichtig, die richtigen beiden Halbbilder zusammenzufügen. Hierbei können Fehler auftreten in der Form, daß die Halbbilder falsch zusammengefügt werden.

### 5.3.1 Die wichtigsten Fehler

1. Die Beschreibung der Halbbilder zu der aufgenommenen Reihenfolge stimmt nicht.

Als Beispiel kann man sich einen Kinofilm in PAL mit 25 *Bildern/s* und zwei Halbbildern je Bild vorstellen. Jedes Bild kann top oder bottom first wiedergegeben werden. Wenn man mit der falschen Ordnung aufnimmt, also mit einem top first statt einem bottom first (und natürlich auch umgekehrt), sind die beiden Halbbilder dann über zwei Bilder verteilt. Den Fehler sieht man auf dem Fernseher nicht, da dort immer

nur ein Halbbild gezeigt wird. Beim Monitor des Computers sieht man das Problem deutlich, da hier immer beide Halbbilder eines Bildes gleichzeitig angezeigt werden.

2. Die Halbbilder sind vertauscht.

Aus einem top Halbbild wird ein bottom Halbbild und umgekehrt. Dies passiert manchmal durch zufällige Fehler, die die TV-Karte nicht kompensieren kann.

3. Bei der Aufnahme oder der Wiedergabe wird die zeitliche Reihenfolge der Halbbilder falsch beschrieben. Dadurch sind u. U. Halbbilder, die zeitlich später folgen sollten, zuerst sichtbar.

### 5.3.2 Fehlererkennung

1. Dieser Fehler ist recht schwer zu erkennen. Manchmal sieht man Szenen, die ohne ersichtlichen Grund flackern. Hier empfiehlt es sich, mit der unten empfohlenen Option neu zu komprimieren. Wenn es danach wesentlich besser ist, war dies der Fehler.
2. Man bemerkt ein Bildzittern und unerklärbare Zeilenfehler. Am besten erkennt man diesen Fehler aber an einem Logo oder einer horizontalen Kante. Die obere oder untere Grenzlinie sieht so aus, als wäre sie losgelöst, was dann etwa so aussehen kann:



Abbildung 5.1: Interlacing Problem

Am Text erkennt man deutlich, daß etwas nicht stimmt. Ob es aber nun ein falsch erkanntes Top/Bottom Field ist oder die Halbbilder nur irgendwo falsch beschriftet wurden, kann man aufgrund eines Bildes aber nicht beurteilen.

3. Dieser Fehler verursacht ein Zittern der Bilder. Zum Erkennen sucht man mit dem Program „glav“ (s.u.) einen Szenenwechsel. Wenn man bei dem Bild, bei dem der Szenenwechsel stattfindet, beim top Halbbild noch das alte Bild sieht und beim bottom Halbbild schon das neue oder umgekehrt, liegt dieser Fehler vor. Es wurde mit der falschen Halbbildreihenfolge aufgenommen.

Wenn man „glav“ oder „lavplay“ (s.u.) zur Fehlersuche benutzt, erkennt man die vorliegenden Probleme nur, wenn man die Option **-F/-flicker** verwendet. Diese Option schaltet einige Optionen, die das Bild verschönern, aus, dafür erkennt man die Probleme leichter.

Will man sich das komprimierte Video anschauen, kann man auch „yuvplay“ (s.u.) verwenden:

```
> lav2yuv | ... | yuvplay
```

Damit man das vorliegende Problem leichter erkennt, schneidet man am besten zuerst einige

Bilder aus dem Video heraus und schaut sie sich an. Das macht man mit folgender Befehlskette:

```
> mkdir pnm
> lav2yuv -f 40 Datei.avi | y4mtopnm | pnmsplit - pnm/Bilde%d.pnm
> rm pnm/Bilde?.pnm
> cd pnm
> xv &
```

Mit dieser Befehlskette wird mit dem ersten Befehl ein Verzeichnis erzeugt, in dem die Bilder gespeichert werden. Mit dem Befehl `lav2yuv -f 40` werden die ersten 40 Bilder auf `stdout` geschrieben. Das Programm „y4mtopnm“ konvertiert die Bilder in das pnm-Format. Das folgende Programm „pnmsplit“ zerlegt jedes Bild in zwei Halbbilder, die einzeln abgespeichert werden. Dann werden die ersten 10 Bilder gelöscht, weil „pnmsplit“ die %0xd Nummerierung nicht unterstützt. Ohne die führende 0 in der Nummerierung werden die Bilder in der falschen Reihenfolge sortiert, was zu einer verwirrenden Wiedergabe führt. Mit dem Lieblingsgrafikprogramm (im Beispiel ist das „xv“) können dann die Bilder betrachtet werden. Da jedes Bild jetzt nur noch ein Halbbild ist, sehen die Bilder gestaucht aus. Wenn man die Bilder betrachtet, sollte man schön langsam vorgehen. Handelt es sich um einen Film, sollte man aufeinander folgend immer zwei fast gleiche Bilder sehen. Fast deshalb, weil das Vollbild auf zwei Halbbilder aufgeteilt wird für die Übertragung. Das bemerkt man recht einfach, wenn beim Anhalten des Films Kammefekte auftreten. Die zusammengehörenden Bilder sollten aufeinander folgen und mit einer geraden Zahl beginnen. Wenn man also die Bilder betrachtet, sollten die Bilder 4 und 5 fast identisch sein, die Bilder 5 und 6 verschieden, 6 und 7 identisch, 7 und 8 verschieden usw. Stimmt diese Reihenfolge nicht, liegt einer der o.a. Fehler vor.

Beim NTSC-Format kann noch ein weiterer Fehler auftreten. Videos in NTSC werden oft im „3:2“ pulldown gesendet. Die Hälfte der Bilder enthält Halbbilder von einem Bild und die andere Hälfte Halbbilder von zwei Bildern. Um diese Uneffektivität für die MPEG Komprimierung zu beseitigen, benötigt man das Programm „yuvkinco“.

Bei einer Quelle, die Halbbilder sendet, wie z. B. eine TV-Kamera, besteht jedes Bild aus zwei Halbbildern, die zu zwei verschiedenen Zeitpunkten aufgenommen wurden und auch hintereinander gezeigt werden. Hierbei auftretende Probleme können schwerer erkennbar sein. Dazu sucht man ein Objekt, das sich im Film waagrecht bewegt, z. B. von links nach rechts. Wenn man sich die extrahierten Halbbilder ansieht, sollte sich das Objekt in kleinen Schritten kontinuierlich bewegen, und nicht einen großen Schritt nach rechts, im folgenden Bild wieder einen kleinen Schritt zurück, usw.

Man darf nicht erwarten, daß die Bildordnung (top oder bottom first) konstant ist. Sie kann sich von Sender zu Sender ändern, von Film zu Film und sogar innerhalb eines Films! Wenn sie sich immer wieder ändert, muß die Kompression durch „mpeg2enc“ mit der Option `-I 1` oder auch `-I 2` vorgenommen werden.

Diese Probleme kann es nur geben, wenn man in voller Größe aufnimmt!

### 5.3.3 Fehlerbehebung

1. Die einzige Möglichkeit, diesen Fehler nach der Aufnahme zu korrigieren ist, die Halbbilder über die Bilder zu verschieben. Diese Problem korrigiert „yuvcorrect“(s.u.) mit der Option `-T BOTT_FORWARD/TOP_FORWARD`. Die zur Zeit vorliegende Halbbildreihenfolge findet man heraus, indem man sich die ersten Zeilen der Bildauslesung

von „lav2yuv“ ansieht:

```
lav2yuv -v 2 Datei.eli >/dev/null
```

lav2yuv gibt bei diesem Kommando eine Zeile am Anfang mit dem interlacing Type: **interlacing: top field first** aus. Hier muß man dann die gegenteilige Option verwenden, in diesem Fall also bei yuvcorrect als Option TOP\_FORWARD verwenden.

Wenn man die Möglichkeit hat, das Video nochmals aufzunehmen, sollte man das mit der umgekehrten Option von -f a oder -f A tun.

2. Die Behebung dieses Fehlers ist einfach mit einem Werkzeug, das die Halbbilder austauschen kann. Dazu kann bei der Komprimierung das Programm „yuvcorrect“ mit der Option -T LINE\_SWITCH eingesetzt werden.
3. Auch die Behebung dieses Fehlers ist einfach, wenn man ein Werkzeug hat, das die Halbbilder beschreibt. Oder man teilt dem Abspielgerät mit, daß es die Halbbilder in einer anderen zeitlichen Reihenfolge abspielen soll, als sie vorliegen. Diese Information gibt man dem Abspielgerät, indem man beim Komprimieren mit dem Programm „mpeg2enc“ (s.u.) die Option -z b/t mitgibt.

Am besten geht man wie folgt vor: man setzt zunächst „yuvcorrect“ mit der Option -T BOTT\_FORWARD oder -T TOP\_FORWARD bei der Komprimierung ein. Sind noch Anfang und Ende der Bilder vertauscht, setzt man „yuvcorrect“ ein zweites mal ein mit der Option -T LINE\_SWITCH, oder man verwendet beim Komprimieren mit dem Programm „mpeg2enc“ die Option -z b/t. Damit man sieht, ob man bei der Fehlerbestimmung richtig lag, werden die Bilder nochmals extrahiert, diesmal aber mit „yuvcorrect“:

```
> lav2yuv -f 40 Datei.avi | yuvcorrect -T OPTION | y4mtopnm |  
pnmsplit - pnm/Bilder%d.pnm
```

Hierin ist OPTION durch die Option zu ersetzen, von der man glaubt, daß sie den Fehler behebt.

Diese Tipps sind für Filmmaterial. Tipps für Material, das von TV-Kameras kommt, werden weiter unten gegeben.

## 6 Videofilme von einzelnen Bildern erzeugen

Mit dem Programm `jpeg2yuv` kann man aus einzelnen Bildern, die im `jpeg`-Format vorliegen, einen `yuv`-Film erzeugen, der dann als Datei gespeichert werden kann, oder gleich mittels des Programms `mpeg2enc` (siehe unten) in ein `mpeg`-Video konvertiert werden kann. Das Speichern des rohen `yuv`-Films erfolgt mit dem Befehl

```
> jpeg2yuv -f 25 -I p -j image%05d.jpg >Ausgabedatei.yuv
```

Dieser Befehl erzeugt eine Datei mit dem Namen `Ausgabedatei.yuv`, die einen Videofilm mit 25 *Bilder/s* enthält. Die Option `-f 25` setzt die Bildrate auf die gewünschten 25 *Bilder/s*. Durch den Namen `image%05d.jpg` werden die Bilder mit den Namen `image00000.jpg`, `image00001.jpg` und so weiter verwendet. (Der Zusatz `05` bedeutet fünf Ziffern, `04` heißt vier Ziffern usw.)

Das direkte Komprimieren der `jpeg`-Bilder ohne Erzeugung einer separaten Datei erfolgt mit dem Befehl

```
> jpeg2yuv -f 25 -I p -j image%05d.jpg | mpeg2enc -o video.m1v
```

Im Gegensatz zum ersten Befehl, mit dem ein `yuv`-Video erzeugt wird, wird hier ein `mpeg`-Video generiert. Die weitere Verwendung des Programms `mpeg2enc` wird weiter unten ausführlich erklärt.

Die Verwendung von weiteren Programmen in der Programmfolge ist möglich. So kann z. B. eine zusätzliche Bildskalierung durch das Programm `yuvscaler` erfolgen. Soll der `mpeg`-Film als `SVCD` Film erzeugt werden, lautet der Befehl

```
> jpeg2yuv -f 25 -I p -j image%05d.jpg | yuvscaler -0 SVCD | mpeg2enc -o video.m1v
```

Manchmal kann es auch notwendig sein, einen Filter zu verwenden. Das erzeugte Video ist dann weicher und etwas weniger scharf.

```
> jpeg2yuv -f 25 -I p -j image%05d.jpg | yuvmedianfilter | mpeg2enc -o video.m1v
```

Ob Filter verwendet werden sollten, hängt von der Qualität der Aufnahme und der gewünschten Kompression ab.

Mit der Option `-b` kann die Startnummer angegeben werden, die das erste Bild der Bildfolge hat. Heißt das erste Bild `image01.jpg` (und nicht `image00.jpg`), dann wird diese Option gesetzt.

```
> jpeg2yuv -b 1 -f 25 -j image*.jpg | yuv2lav -o VideooohneTon.avi
```

Der Ton kann einfach durch den Befehl

```
> lavaddwav VideooohneTon.avi Tonspur.wav VideomitTon.avi.
```

hinzugefügt werden. Hier wird dem Video die Tonspur Tonspur.wav hinzugefügt. Die Anzahl der Bilder, die verwendet werden sollen, kann mit der Option -n bestimmt werden. Für Bilder im ppm-Format gibt es das Programm ppmttoy4m, zu dem es auch eine manpage gibt. Der Befehl zum Erzeugen eines Videos aus Bildern im ppm-Format sieht so aus:

```
> cat *.ppm | ppmttoy4m -o 75 -n 60 -F 25:1 | mpeg2enc -o Ausgabevideo.m1v
```

Hier werden alle Bilder im ppm-Format durch den Befehl cat an das Programm ppmttoy4m weitergeleitet. Dieses Programm ignoriert durch die Option -o 75 die ersten 75 Bilder. Mit -n 60 werden die folgenden 60 Bilder an das Programm mpeg2enc weitergeleitet, das die Bilder komprimiert. Die Option -F 25:1 bestimmt die Bildfolgefrequenz, die für PAL-Filme gesetzt werden muß, da die Voreinstellung (default) für NTSC-Filme (30 000:1 001) gilt. Es können auch andere Bildformate verwendet werden, wenn man sie vor der Verarbeitung in das ppm-Format konvertiert. Dies kann z. B. so aussehen:

```
> ls *.tga | xargs -n 1 tgatoppm | ppmttoy4m | yuvplay
```

Hier wird eine Liste von Bildnamen (ls \*.tga) an das Programm xargs übergeben. xargs startet das Programm tgatoppm mit einem Argument (-n 1) per Aufruf und leitet die Ausgabe an das Programm ppmttoy4m weiter. Diesmal wird das Video nur am Bildschirm angezeigt. Das Programm xargs wird nur benötigt, wenn der Konverter (tgatoppm) nur ein Bild pro Aufruf verarbeiten kann.

Wenn die Konvertierung mit dem Programm ImageMagick (dem Schweizer Armeemesser unter den Konvertierungsprogrammen) erfolgen soll, lautet der Befehl

```
> convert *.jpg ppm:- | ppmttoy4m | yuvplay
```

Hier werden alle \*.jpg Dateien im aktuellen Verzeichnis ins ppm-Format konvertiert und auf stout geschrieben. Dort wartet schon ppmttoy4m und bearbeitet die Bilder weiter...

# 7 Abspielen des aufgenommenen Films

Zur Wiedergabe der Rohaufnahmen dienen die beiden Programme „lavplay“ und „glav“. Beide Programme nutzen weitgehend die gleichen Optionen, so daß diese nur einmal aufgeführt werden. Während das Programm „lavplay“ ein reines Kommandozeilenprogramm ist, bietet „glav“ ein GUI. „glav“ kann auch zur Bearbeitung des aufgenommenen Materials verwendet werden (siehe nächstes Kapitel). Die Wiedergabe mittels Software playback erfolgt mit dem Befehl

```
> lavplay -p S Datei.avi
```

Man sollte das aufgenommene Video sehen und den Ton hören. Soll die Wiedergabe ohne Ton erfolgen, muß zusätzlich die Option -a 0 gesetzt werden.

```
> lavplay -p S -a 0 Datei.avi
```

Bei dieser Art der Wiedergabe ist die CPU hoch belastet, da die ganze Decodierung softwaremäßig geschieht. Man benötigt aber sonst nichts für die Wiedergabe.

Bei ZORAN-Chip basierten Karten ist die hardwaremäßige Decodierung der bessere Weg:

```
> lavplay -p H Datei.avi
```

Hierbei ist die CPU-Last sehr niedrig.  
Eine weitere Möglichkeit ist der Befehl

```
> lavplay -p C Datei.avi
```

Hier wird für die Ausgabe auch die Hardware verwendet, aber die Ausgabe erfolgt auf dem TV-Ausgang der Karte. Bei allen Befehlen kann „lavplay“ durch „glav“ ersetzt werden. Dann hat man zusätzlich zum Wiedergabebild ein GUI. Bei Hardware Wiedergabe wird das Signal für Composite und SVHS Out erzeugt; man kann das aufgenommene Video auch am Fernsehen sehen.

Eine alternative Möglichkeit, die Aufnahmen anzuschauen, ist das Programm „yuvplay“ . Hier erfolgt allerdings keine Tonwiedergabe.

```
> lav2yuv Datei.avi | yuvplay
```

Mit yuvplay im Kompressionskommando sieht man die Änderungen, die Filter und Scaler machen. Man kann das auch für Zeitlupen verwenden.

Hier noch einige wichtige Hinweise:

- Versuchen Sie nie, xawtv und lavplay oder glav mit Hardwareplayback nebeneinander auf einer Zoran Karte laufen zu lassen, das funktioniert nicht, Softwareplayback und xawtv nebeneinander hingegen schon.
- Sobald der Treiber geladen wurde, muß man mindestens einmal xawtv starten. Das setzt einige Dinge, die lavplay und glav nicht machen, aber benötigt werden für ein Hardwareplayback. Nicht vergessen, xawtv wieder zu schließen!
- Versuchen Sie nie, glav oder lavplay in den Hintergrund zu schicken, das funktioniert nicht!
- SECAM Playback funktioniert derzeit (12. 3. 2001) nur in monochrom, aber die Aufnahme und die Komprimierung werden richtig gemacht.
- Schlechte Kabel beeinträchtigen die Qualität. Man bemerkt das besonders bei Buchstaben oder scharfen Kanten: sieht man dort Schatten, sollte man ein besseres Kabel verwenden.



# 8 Bearbeitung des Videos

## 8.1 Bearbeiten mit glav

Die meisten Aufgaben bei der Bearbeitung eines Videos kann man mit „glav“ erledigen. Man kann hiermit Teile aus dem Video löschen, herausschneiden, kopieren und Teile einfügen. Startet man „glav“ z. B. mit dem Befehl

```
> glav -p S Datei.avi
```

oder

```
> glav -p S Datei0*.avi, (wenn mehrere avi-Dateien zu einem Film gehören)
```

öffnet sich das in der Abbildung gezeigte GUI.

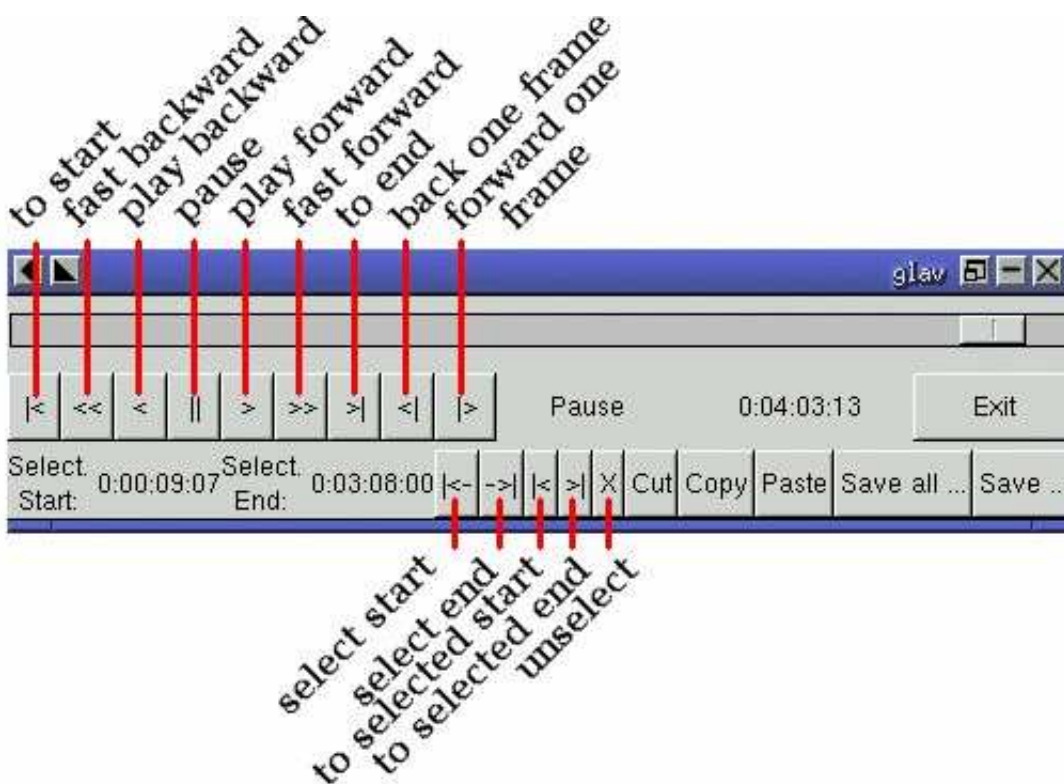


Abbildung 8.1: glav GUI

Soll die Bearbeitung ohne Tonwiedergabe erfolgen, ist zusätzlich die Option **-a 0** zu setzen. Glav verändert bei der Bearbeitung des Films die Quelldateien nicht. Die gemachten Ände-

rungen werden in einer separaten Datei mit der Endung Datei.eli (für edit list) abgespeichert. Diese Edit-List-Datei ist eine Textdatei, die man auch mit einem Editor bearbeiten kann. Mit dieser Edit-List-Datei können alle lavtools umgehen wie z. B. lav2wav, lav2yuv, lavtrans, ...

Das Programm ist weitgehend selbsterklärend. Wenn man den Anfang und das Ende eines Videos wegschneiden will, markiert man mit select start und select end den gewünschten Anfangs- und Endpunkt und speichert dann mit save select als Datei.eli ab. Die so gespeicherte Edit List Datei wird dann als Eingabedatei für die lavtools verwendet. Wenn man ein aufgenommenes Video in mehrere kleine Teile zerlegen will, wiederholt man diesen Vorgang einfach mehrmals. Dabei darf nicht vergessen werden, jeweils andere Namen für die abgespeicherten Teile zu verwenden.

Die Änderungen werden sofort durchgeführt und sind sofort verfügbar, ohne daß etwas berechnet werden muß.

Wenn man einen avi-file hat, aus dem man mit glav eine Kurzversion in eine Datei.eli abgespeichert hat und man möchte daraus ein eigenständiges avi-file haben, kann das so realisiert werden:

```
> lavtrans -o Kurzversion.avi -f a Datei.eli
```

Es bedeuten:

**-o Kurzversion.avi**

Name der neuen Kurzversion

**-f a**

Ausgabeformat, hier avi

**Datei.eli**

Quelldatei, in der die mit glav gemachten Änderungen abgespeichert sind.

## 8.2 Mehrere Videos zu einem vereinen

Will man aus mehreren einzelnen Videos ein einziges Video herstellen, geht das mit folgendem Befehl:

```
> lavtrans -o NeueDatei.qt -f q Datei1.avi Datei2.avi ...
```

Es bedeuten:

**-NeueDatei.qt**

Name der Datei mit den vereinigten Videos

**-f q**

Ausgabeformat, im Beispiel Quicktime

**Datei1.avi Datei2.avi ...**

Zu vereinigende Videos

Videos werden nur sehr selten zusammengefügt. Es ist ausserdem wichtig, daß bei den Videos alle Parameter gleich sind, wie etwa Interlacing, Bildgrösse usw, sonst funktioniert es nicht. Bei diesem Zusammenfügen von Videos darf nicht vergessen werden, daß es da noch das 2 GB Filegrößenlimit auf 32 Bit Systemen mit älterer glibc gibt, das bei AVI- und Quicktime-Formaten die Filegröße begrenzt. AVI ist bei den mjpegtools sowieso immer nur 2 GB groß.

## 8.3 Ton auskoppeln

Will man die Tonspur eines Videos ohne Bilder auskoppeln und daraus eine wav-Datei zur weiteren Verwendung machen, bieten die lavtools dazu zwei Möglichkeiten.

```
>lavtrans -o Tonspur.wav -f w Datei.avi
```

Mit diesem Befehl wird die Tonspur des Videos mit dem Namen Datei.avi ausgelesen und in eine wav-Datei mit dem Namen Tonspur.wav gewandelt. Dies kann sinnvoll sein, wenn man Rauschen entfernen will oder wenn man die wav-Datei in ein anders Format (z.B mp3) konvertieren will für weitere Verwendung.

Die zweite Möglichkeit ist

```
> lav2wav Datei.avi >Tonspur.wav
```

Die zweite Möglichkeit hat den Vorteil, daß man die Wav-Datei direkt mit einem anderen Programm verarbeiten kann (Pipes!), statt einen wav-file anzulegen.

## 8.4 Einzelbilder aus dem Video auskoppeln

Zum Auslesen einzelner Bilder aus einem Video wird zunächst ein Verzeichnis (hier: jpg) erzeugt, in dem die Bilder abgespeichert werden.

```
> mkdir jpg
```

Danach werden mit dem Befehl

```
> lavtrans -o jpg/image%05d.jpg -f i Datei.avi
```

die einzelnen Bilder als image00000.jpg, image00001.jpg, ... im Verzeichnis jpg/ abgespeichert.

Handelt es sich bei dem Quellvideo Datei.avi um einen interlaced Film, so werden die Halbbilder hintereinander im File abgespeichert. Da viele Grafikprogramme nur das erste Bild im File anzeigen, sieht man nur ein Halbbild. Es sollten in diesem Fall zwei aufeinanderfolgende Bilder zu einem Vollbild zusammengefügt werden. Ein vollständiges Bild abspeichern geht mit dem Befehl

```
> lav2yuv -f 1 Datei.avi | y4mtoppm -L Bild.pnm
```

Wenn man die Halbbilder in einzelne Dateien speichern will, geht das wie folgt:

```
> lav2yuv -f 5 ../Datei.avi > y4mtoppm > pnmsplit - image%d.pnm
```

Dies ist möglicherweise sinnvoll, wenn man einzelne Bilder benötigt und sich nicht mit dem Suchen eines einzelnen Bildes beschäftigen will.

## 8.5 Überblendungen erzeugen

Dank der lavpipe von Philipp Zabel ist es nun möglich, einfache Überblendungen zwischen Filmen oder Kombinationen mehrerer Bilder zu machen.

Philipp schrieb für die Herstellung von Überblendungen folgendes HOWTO:

Lassen Sie uns einfach folgendes Szenarium vorstellen: Wir haben zwei Eingangsvideos mit den Namen intro.avi und epilogue.mov. Wir möchten nun von intro.avi in epilogue.avi in der Zeit von einer Sekunde überblenden (das bedeutet 25 Bilder in PAL oder 30 Bilder in NTSC).

Intro.avi und epilogue.mov müssen dazu das gleiche Format haben (gleiche Bildfolgefrequenz und gleiche Auflösung). In diesem Beispiel sind beide Videos 352x288 PAL Filme. Intro.avi besteht aus 250 Bildern und epilogue.mov ist 1 000 Bilder lang.

Deshalb wird unser Ausgabefilm enthalten:

- die ersten 225 Bilder von intro.avi
- eine 25 Bilder dauernde Überblendung, die die letzten 25 Bilder von intro.avi und die ersten 25 Bilder von epilogue.mov enthält
- die letzten 975 Bilder von epilogue.mov.

Wir bekommen die letzten 25 Bilder von intro.avi, indem wir eingeben:

```
> lav2yuv -o 225 -f 25 intro.avi
```

Hierin bedeuten

### **-o 225**

Das ist der Offset. Er sagt dem Programm lav2yuv, daß es bei Bild 225 beginnen soll.

### **-f 25**

Erstellt einen 25 Bilder langen Ausgabefile nach dem Offset.

Eine andere Möglichkeit dazu ist folgende

```
> lav2yuv -o -25 intro.avi
```

Das funktioniert, weil negative Offsets vom Ende an gezählt werden.

Die ersten 25 Bilder von epilogue.mov erhält man mit

```
> lav2yuv -f 25 epilogue.mov
```

-o fehlt, weil die Voreinstellung des Offsets null ist.

Jetzt müssen die beiden Filme mit dem Programm lavpipe kombiniert werden. Der Befehl dazu lautet

```
> lavpipe 'lav2yuv -o 225 -f 25 intro.avi' 'lav2yuv -f 25 epilogue.mov'
```

Die Ausgabe ist ein roher yuv Film, der in transist.flr eingegeben werden kann.

Transist.flt muß über die Dauer der Überblendung und die Undurchsichtigkeit des zweiten Films zu Beginn und am Ende der Überblendung informiert werden. Dies geschieht durch folgende Optionen:

**-o num**

Bezeichnet die Dauer der Undurchsichtigkeit des zweiten Eingangs zu Beginn [0-255]

**-O num**

Bezeichnet die Undurchsichtigkeit des zweiten Eingangs am Ende [0-255]

**-d num**

Bezeichnet die Dauer der Überblendung in Bilderanzahl

Ein Undurchsichtigkeit von null bedeutet, daß der zweite Film vollkommen transparent ist (nur der erste Film ist sichtbar), bei 255 ist der zweite Film vollkommen sichtbar.

In unserem Fall bei der Überblendung des Films 1 in den Film 2 lautet der korrekte Aufruf:

```
> transist.flt -o 0 -O 255 -d 25
```

Die Optionen **-s** und **-n** entsprechen den Optionen **-o** und **-f** bei dem Programm lav2yuv. Sie werden nur benötigt, falls jemand nur einen Teil der Überblendung wiedergeben möchte oder aus welchem Grund auch immer. Bitte beachten Sie, daß dies nur die wichtenden Rechnungen beeinflusst, keiner der Eingänge wird wirklich übersprungen. Wenn Sie die Übersprungsoption benutzen (**-s 30** z. B.), müssen Sie auch die ersten 30 Bilder in lav2yuv (**-o 30**) überspringen, um das erwartete Ergebnis zu erzielen. Wenn Sie dies nicht verstehen, senden Sie eine email an den Autor oder ignorieren Sie einfach die Optionen **-s** und **-n**. Der ganze Vorgang wird eventuell automatisiert.

Nun möchten wir den Film komprimieren mit yuv2lav:

```
> yuv2lav -f a -q 80 -o transition.avi
```

Mit diesem Befehl wird der yuv Film vom Standardinput gelesen und ein avi File ausgegeben mit einer jpeg Kompressionsqualität von 80 %.

Jetzt kennen wir das gesamte Kommando um eine Überblendung zu erstellen:

```
> ypipe \lav2yuv -o 255 -f 25 intro.avi\ \lav2yuv -f25 epilogue.mov\ | transist.flt  
-o 0 -O 255 -d 25 | yuv2lav -f a -q 80 -o transition.avi
```

Das entstehende Video kann als LAV Edit List geschrieben werden, ein Textfile, der die folgenden Zeilen enthält:

```
LAV Edit List  
PAL  
3  
intro.avi  
epilogue.mov  
0 0 224  
1 0 24  
2 25 999
```

Dieses File dient als Eingabedatei für glav oder lavplay, oder Sie bauen es in eine Pipe zusammen mit mpeg2enc mit lav2yuv ein oder den ganzen Kram in einem einzelnen mpeg file mit lavtrans oder lav2yuv | yuv2lav kombinieren.

# 9 Einen Film in MPEG- oder DIVx Videos umwandeln

## 9.1 Allgemeines

Da die Rohdaten im Aufnahmeformat sehr viel Speicherplatz benötigen, ist es notwendig, die Bildfolgen zu komprimieren, damit sie auf die herkömmlichen Speichermedien wie DVD und CD passen. Dazu werden zuerst Ton und Bild getrennt von den Rohdaten ausgelesen. Dies geschieht mit den Programmen `lav2wav` und `lav2yuv`. Diese beiden Datenfiles werden dann jeder für sich komprimiert. Der Tonfile wird in dem Format `mp2` und der Bildteil in dem Format `mpg` erstellt. Anschließend werden die beiden komprimierten files wieder zusammengeführt. Die Kompression erfolgt für den Ton mit dem Programm `mp2enc` und für das Bild mit `mpeg2enc`. Es ist dabei sinnvoll, die Befehle in einer sogenannten pipe aufzuschreiben oder sie in einem Script zusammenzufassen. Ein solches Script ist `lav2mpeg`, mit dem man in einem Kommando ein `mpg`-Video erzeugen kann.

Der Befehl für die Komprimierung mit dem `lav2mpeg` Script sieht für das `mpeg1` Ausgabeformat z. B. so aus:

```
> lav2mpeg -a 160 -b 2110 -d 320x240 -m mpeg1 -o Ausgabedatei.mpg Datei.eli
```

Hierin bedeuten die einzelnen Optionen

- **-m mpeg1:** Das Ausgabeformat ist `mpeg1`
- **-a 160:** Die Audiobitrate beträgt  $160 \frac{kBit}{s}$
- **-b 2110:** Die Videobitrate ist  $2110 \frac{kBit}{s}$
- **-d 320x240:** Die Bildgröße beträgt `320x240`

Für die Erzeugung eines `mpg2`-files lautet der Befehl

```
> lav2mpeg -o mpeg2 -0 Ausgabedatei.mpg Datei.eli
```

- Dies erzeugt eine `mpeg2` Datei mit der voreingestellten Bitrate in der gleichen Größe wie die Eingangsdatei.

Bessere Ergebnisse kann man jedoch erzielen, indem man die verschiedenen Optionen ausprobiert und so herausfindet, welche auf dem vorliegenden Rechner die qualitativ besten Ergebnisse ergeben. Die möglichen Optionen werden unten im einzelnen vorgestellt.

Die meisten Optionen werden an Beispielen für das `mpeg1`-Format erklärt, weil die meisten von Ihnen auch für die Erzeugung von anderen Ausgabeformaten gültig sind.

Wie man bei der Erzeugung von VCD/SVCD Stills Sequenzen erzeugt (Optionen **-f 6** und

-f 7 im Komprimierungsprogramm mpeg2enc) kann bei <http://www.mir.com/DMG/> nachgelesen werden. Stills sequences werden für die Erzeugung von Menus bei VCD/SVCD benötigt

## 9.2 Grundsätzliches zur Tonkomprimierung

Mpeg1 Videos benötigen den Ton im mp2 Format. Für mpeg2 Videos kann der Ton im mp2 oder mp3 Format sein. Das mp3 Format ist kein offiziell gültiges Format für DVDs, aber manche DVD Spieler können es abspielen. Man sollte das mp2 Format verwenden, weil die meisten mpeg2 Abspielgeräte keine mpeg2 Videos mit Ton im mp3 Format spielen können (DVD Spieler, speziell die verschiedenen Winxx Versionen, haben hiermit große Probleme). Das Programm mp2enc ist ein mp2 Audio Encoder. Der Encoder toolame kann auch mp2-files erzeugen, so daß man auch ihn anstelle von mp2enc verwenden kann. Toolame ist schneller als mp2enc, aber er unterstützt kein resampling (zB: 48 000 nach 44 100  $\frac{\text{samples}}{s}$ ). Viele Abspielgeräte spielen SVCDs mit einer Audiorate von 48 000  $\frac{\text{samples}}{s}$  ab. Für die Erzeugung von mp3-files haben Sie sicher ein Kompressionsprogramm (z. B. lame).

Ein Beispiel für das Auslesen des Tons von Rohdatenfiles und Komprimierung ins mp2 Format ist folgendes:

```
> lav2wav Datei.avi | mp2enc -o Ausgabetondatei.mp2
```

Diese Befehlsfolge erzeugt eine mp2-Tondatei aus dem Rohdatenfilm Datei.avi mit einer Bitrate von 224  $\frac{k\text{Bit}}{s}$  und einer sampling rate von 44.1kHz. Die Bitrate sollte für Hardware Playback verwendet werden. Man kann natürlich mehrere Dateien angeben oder den Platzhalter %02d verwenden. Wenn man ein Tondatei mit 44.1kHz hat wir der Ton auf 48kHz hinaufgesampelt. Wenn man eine Sempel Rate von 44.1kHz haben will so muss man die Option -r 44100 zu dem mp2enc Befehl hinzufügen.

Ein weiteres Beispiel für die Tonkomprimierung ist folgender, wenn der Ton schon als wav Datei vorliegt:

```
> cat Tondatei.wav | mp2enc -v 2 -V -o Ausgabetondatei.mp2
```

Dieser Befehl erzeugt erzeugt eine VCD kompatible (-V bitrate=224, Stereo, sampling rate 44 100) Tondatei aus einer wav Datei. Die Option -v 2 veranlaßt das Programm mp2enc, mehr Informationen während der Komprimierung auszugeben. Man sieht die Anzahl der bereits komprimierten Sekunden.

Ob die Kompression erfolgreich war kann mit

```
> plaympeg Ausgabetondatei.mp2
```

getestet werden. Statt dem Abspielprogramm plaympeg können auch andere Programme wie z. B. mpg123 oder xmms zum Abspielen verwendet werden.

Wenn man nicht vorhat, die komprimierten Videos auf einem separaten Abspielgerät abzuspielen, kann man für den Ton auch das mp3 Format verwenden, was viele Softwareabspieler lesen können.



## 9.3 Grundsätzliches zur Videokomprimierung

Mit den **lavtools** können mpeg1 und mpeg2 Videos erzeugt werden.

Normalerweise ist das erste Video, das man erzeugt, qualitativ nicht das beste, da man ja noch nicht weiß welche der verschiedenen möglichen Optionen für das vorliegende Video die besten sind. Um das optimale Verhältnis von Qualität und Größe zu erhalten, sollte mit den Optionen `bitrate`, `search radius`, `noise filter` und weiteren gespielt werden. Alle erlaubten Optionen sind in der manpage des Programms `mpeg2enc` erklärt.

Beispiel für die Bildkomprimierung:

```
> lav2yuv Datei.avi | mpeg2enc -o Ausgabevideo.m1v
```

Mit diesem Befehl wird ein Video mit den Voreinstellungswerten des Programms erstellt. Die Bitrate beträgt dabei  $1152 \frac{kBit}{s}$ . Dies ist genau die Bitrate, die für VCDs benötigt wird. Ein weiteres Beispiel, bei dem nicht die Voreinstellungswerte benutzt werden, ist

```
> lav2yuv Datei*.avi | mpeg2enc -b 1500 -r 16 -o Ausgabevideo.m1v
```

Hiermit erzeugt `mpeg2enc` ein Video mit einer Bitrate von  $1500 \frac{kBit}{s}$  und verwendet dabei einen Suchradius von 16. Das bedeutet, daß der Bildkomprimierer, wenn er versucht, ähnliche Macroblocks von Pixeln zwischen zwei Bildern zu finden, bis zu 16 Pixel entfernt vom Ausgangspunkt sucht. Er sucht zweimal so weit, wenn er von Bild zu Bild vergleicht. Vernünftige Werte für den Suchradius sind 16 und 24. Die Voreinstellung ist 16, so daß  $\frac{1}{2}$  die Option in obigem Befehl eigentlich berflüssig ist. Kleinere Werte (0;8) erhöhen die Kompressionsgeschwindigkeit, aber man erhält schlechtere Qualität (mehr sichtbare Artefakte), größere Werte (24;32) verbessern die Qualität zu Lasten der Geschwindigkeit. Durch die Angabe von \* im Dateinamen werden alle Dateien verarbeitet, die den Namen Datei\*.avi (Datei01.avi, Datei02.avi, Dateiabcd.avi, Datei.avi ··· ) haben.

Hier noch einige wichtige Hinweise für die Bildkompression:

- Die hier gezeigten ersten Beispiele decken nur einen kleinen Teil der möglichen Optionen ab, von denen viele zusammen nutzbar sind, um gute Qualität bei kleinen Bitraten zu erhalten.
- Die Bildkomprimierung nimmt sehr viel Zeit in Anspruch. Die benötigte Zeit steigt mit größerem Suchradius und üblicherweise mit der Anzahl der verwendeten Optionen.
- Bei mpeg1 ist VBR (variable bit rate) nicht default (keine Voreinstellung) aktiviert. Wenn man nun mit der Option **-q 15** (siehe weiter unten) komprimiert, setzt `mpeg2enc` die maximale Bitrate auf 1152 ohne Angabe der Bitrate. Wenn die gewünschte Qualität nicht mit der eingestellten Bandbreite erreicht wird, verschlechtert `mpeg2enc` einfach die Bildqualität. Wenn man nun ein mpeg1 Video mit VBR haben will, sollte man die Bitrate höher (z. B 2500) setzen, sonst macht es kaum Sinn. VBR mpeg1 Videos werden von vielen Abspielgeräten nicht akzeptiert.
- Um während des Lernprozesses mit der Suche der optimalen Optionskombination die Fortschritte (oder Rckschritte) besser einordnen zu können, empfiehlt es sich, den Namen des Ausgabevideos um die verwendeten Optionen zu erweitern (z. B Ausgabevideo\_b1500\_r16\_41.21.m1v). Eine weitere Möglichkeit, die Dateien besser unterscheiden zu können ist, für Tondateien konsequent die Endung `.mp2` bei layer 2 Audio, für

mpeg1-Videos die Endung `.m1v` und für mpeg2-Videos die Endung `.m2v` zu verwenden. Normalerweise reserviert man die Endung `.mpg` für den vollständigen Film, bei dem Ton und Bild wieder zusammengeführt sind.

### 9.3.1 Skalieren

Mit dem Programm `yuvscaler` kann man die Bildgröße des Videos vor der Kompression ändern. Dies ist sehr nützlich, wenn man Besitzer von DC10 und DC10+ Karten ist, da diese mit `-d 1 768x576` oder `-d 2 384x288` (PAL/SECAM) oder `-d 1 720x480` (NTSC) `-d 2 352x240` (NTSC) aufnehmen.

Eine Beschreibung aller Optionen dieses Programms gibt es mit

```
> yuvscaler -h.
```

Ein erstes Beispiel für die Bildskalierung ist der Befehl

```
> lav2yuv Datei.avi | yuvscaler -O VCD | mpeg2enc -o Ausgabevideo.m1v
```

Hiermit wird die Größe der Bilder auf VCD Größe skaliert, d.h. bei PAL/SECAM auf 352x288 und bei NTSC auf 352x240. Das skalierte yuv Video wird anschließend mit `mpeg2enc` als mpeg1 Video komprimiert.

Skalierung auf SVCD Größe ist natürlich auch möglich, also Skalierung auf die Größen 480x480 (NTSC) oder 480x567 (PAL/SECAM). Die zugehörige Option beim `yuvscaler` ist `-O SVCD`.

```
> lav2yuv Datei.avi | yuvscaler -O SVCD -M BICUBIC | mpeg2enc -o Ausgabevideo.m2v
```

Mit der Option `-M BICUBIC` (Mode Key Option) wird `yuvscaler` gezwungen, den qualitativ höherwertigen bicubic Algorithmus für das Downscaling zu verwenden anstelle des voreingestellten resample Algorithmusses. Beim Upscaling wird immer der bicubic Algorithmus verwendet.

Ein weiteres Beispiel ist

```
> lav2yuv Datei.avi | yuvscaler -I USE_450x340+20+30 -O SIZE_320x200 | mpeg2enc -o Ausgabevideo.m1v
```

Mit der Option `-I` wird hier der Bereich des Bildes angegeben, der beim Skalieren benutzt wird. Man kann also nur einen Teil des Bildes verwenden. Die Option `-O` gibt an, auf welche Bildgröße die Ausgabedatei skaliert wird.

`yuvscaler` kann auch einen Teil der Bildes als aktive Fläche weitergeben. Der Rest wird auf schwarz gesetzt, wenn man die Option

**`-I ACTIVE_WidthxHeight+Widthoffset+Heightoffset`**

verwendet. Das Ergebnis kann mit

```
> plaympeg Ausgabevideo.mpg
```

getestet werden.

## 9.4 Grundsätzliches zum Zusammenführen von komprimiertem Ton und Bild

Die komprimierten Ton- und Videodateien müssen nun wieder zu einem Film zusammengefügt werden. Dieser Vorgang wird in der Fachsprache multiplexen oder kurz muxen genannt. Dies erfolgt mit dem Befehl

```
> mplex Ausgabedatei.mp2 Ausgabevideo.m1v -o MeinVideo.mpg
```

Ton und Bild werden zum Videofilm MeinVideo.mpg zusammengefügt. Es kann jetzt mit Ihrem bevorzugten mpeg-Spieler angeschaut werden. Alle SMPG Library basierten Spieler (z. B. gtv) sollten gut funktionieren. Andere Spieler, die gut funktionieren, sind z. B. xine, MPlayer, kaffeine, xmovie.

Nun noch einige wichtige Hinweise:

- Wenn man beim Komprimieren die Option **-S** angegeben hat, wird das Programm mplex den Film automatisch in mehrere Einheiten zerteilen, wenn man beim Ausgabedateinamen den Platzhalter `%d` einfügt (etwa `-o Ausgabedatei%d.mpg`). Die Videos, die so erzeugt werden, sind stand-alone mpeg Videos. Die Größe der einzelnen Videos wird durch die hinter der Option **-S** angegebene Zahl in *MByte* bei `mpeg2enc` bestimmt (z. B. `-S 650` zerlegt den Film in Stücke von *650MB* Größe, so daß sie auf eine normale CD passen).
- Anscheinend hat der Abspieler `xine` ein Problem mit dem Suchen im Video. Der Abspieler `MPlayer` hat Probleme beim „Vorwärts/Rckwärts Suchen“ bei Filmen mit variabler Bitrate, weil er immer im Film die Strecke vorwärts geht, die sich bei einem Film mit fester Bitrate ergäbe. Diese Strecke kann signifikant mehr als 10 Sekunden oder 1 Minute (dies sind die Zeitsprünge, die der `MPlayer` bei jedem Druck auf die Pfeiltasten im Film springt) sein. Wundern Sie sich also nicht, wenn viel mehr Zeit vorwärts gesprungen wird, als Sie erwarten.
- Multiplexen mit variabler Bitrate  
Sie dürfen nicht vergessen, dem Programm `mplex` mitzuteilen, daß ein VBR Film zu verarbeiten ist (dies geschieht mit der Option **-V**), wie dies auch vorher dem Programm `mpeg2enc` mitgeteilt werden muß. Prinzipiell könnte dies automatisch erfolgen, was aber noch nicht funktioniert. Wenn im Komprimierungsprogramm `mpeg2enc` eine Videopuffergröße **-V** angegeben wurde, muß auch beim Programm `mplex` ein Puffer angegeben werden, der mindestens so groß ist wie der bei `mpeg2enc` angegebene. Bei `mpeg1`-Filmen sollte die Videobitrate den Wert  $2800 \frac{kB}{s}$  nicht übersteigen. Die Puffergröße sollte maximal  $400 kB$  betragen und für die Option **-q** (siehe unten) sollte keine bessere Qualität als 6 gewählt werden.

Hierzu ein Beispiel:

```
> mplex -V -r 1740 audio.mp2 video_vbr.m1v -o vbr_film.mpg
```

Hier wird ein Film mit variabler Bitrate (VBR) gemultiplext (zusammengefügt). Da das Programm `mplex` den Film vor dem Zusammenfügen nicht lesen kann (single pass), kann es auch die maximale Bitrate nicht selbst ermitteln. Daher muß sie mit der Option **-r** im Befehl angegeben werden. Die maximale Bitrate des erzeugten Films setzt sich zusammen aus der Audiobitrate, dem Maximum bei der Videobitrate zusätzlich 1 - 2

% für die Multiplexinformation. Wenn eine Audiodatei eine Bitrate von  $224 \frac{kBit}{s}$  hat (-b 224) und die Bilddatei mit der Option **-b 1500 -q 9** komprimiert wurde, ergibt das eine maximale Bitrate von  $224 \frac{kBit}{s} + 1500 \frac{kBit}{s} = 1724 \frac{kBit}{s}$ . Hierzu sind noch die 1-2 % für die Multiplexinformation zu addieren, so daß sich eine maximale Bitrate von  $1740 \frac{kBit}{s}$  ergibt, die in der Option mit **-r 1740** anzugeben ist.

Der so erzeugten Videofilm kann mit

```
> plaympeg vbr_film.mpg
```

oder mit

```
> gtv vbr_film.mpg
```

angeschaut werden.

## 9.5 Videofilme im mpeg1-Format erzeugen

Für Filme im mpeg1-Format ist das Audioformat mp2 und das Videoformat mpeg1 zu verwenden. Eine in den Möglichkeiten des mpeg1-Formats eingeschränkte Version ist das Format VCD. VCDs sind mit konstanter Bitrate (CBR) zu codieren. Bei mpeg1-Formaten, die nur auf Computern abgespielt werden, kann man auch variable Bitrate verwenden. Das Audioformat muß aber immer eine konstante Bitrate haben.

Mpeg1 wird bis zu einer Bildgröße 352x288 bei PAL und 352x240 bei NTSC empfohlen. Bei größeren Bildgrößen ist dann das mpeg2-Format die bessere Wahl. Es gibt aber keine exakte Grenze, bis zu der das mpeg1-Format besser ist als das mpeg2-Format.

### 9.5.1 Beispiele für eine Audiodatei im mpeg1-Format

```
> lav2wav Datei.avi | mp2enc -r 44100 -o Ausgabedatei.mp2
```

Dies funktioniert für mpeg1-Filme ohne Probleme. Man kann noch etwas Platz sparen, wenn man eine kleinere Bitrate (Option **-b**) verwendet. Brauchbare Qualität liefern auch noch  $160 \frac{kBit}{s}$  oder  $192 \frac{kBit}{s}$ . Die **-r 44100** Option bewirkt das die Sampling Rate des files auf jeden Fall die angegebenen 44.1kHz beträgt.

```
> lav2wav Datei.eli | mp2enc -b 128 -m -o Ausgabedatei.mp2
```

Dieser Befehl erzeugt eine mono Ausgabedatei (Option **-m**) mit einer konstanten Bitrate von  $128 \frac{kBit}{s}$ . Als Quelldatei wird diesmal ein Editlistfile verwendet (kann einen beliebigen Namen haben), die aus der Bearbeitung des avi-Rohdatenfilms mit dem Programm glav erzeugt wurde. Alle Änderungen, die mit dem Programm glav an dem Rohdatenfilm gemacht wurden, werden direkt verarbeitet und an das Programm mp2enc weitergeleitet. Man muß also **nicht** den bearbeiteten Film mit dem Programm lavtrans bearbeiten, damit er konvertierbar wird.

## 9.5.2 Beispiele für eine Videodatei im mpeg1-Format

```
> lav2yuv Datei.eli | mpeg2enc -b 2000 -r 24 -q 6 -o Videoausgabedatei.m1v
```

Hier wird ein Film mit einer maximalen Bitrate von  $2000 \frac{kBit}{s}$  mit einem Suchradius von 24 erzeugt. Die Option **-q 6** erzeugt einen Film mit einer variablen Bitrate mit einem Qualitätsfaktor von 6. Als Quelldatei wird auch hier ein mit glav erstelltes Editlistfile verwendet.

**Erklärung:** Wenn das Programm mpeg2enc ohne die Option **-q** aufgerufen wird, erzeugt es einen mpeg-Film mit konstanter Bitrate (CBR). Grob ausgedrückt wird dadurch der Kompressionsfaktor der Bilder (und damit die Bildqualität) so gestaltet, daß jedes Bild die gleiche Größe, also gleich viele Bits, hat. Solche Filme mit konstanter Bitrate braucht man für Fernsehübertragungen und werden für Billighardware wie DVD und VCD verwendet.

Dies ist natürlich recht ineffizient, da bei ruhigen Szenen Bits gespart werden können, die man dann bei schnellen Sequenzen verwendet. Insgesamt wird durch die variable Bitrate Speicherplatz gespart. Wenn nun beim Programm mpeg2enc die Option **-q** verwendet wird, wird ein Film mit variabler Bitrate erzeugt. Wird zusätzlich zur Option **-q** noch eine Bitrate mit der Option **-b** angegeben, so bezeichnet jetzt die Bitrate der **-b** Option die maximal zulässige Bitrate. Gibt man keine Bitrate an, wird die default Bitrate verwendet, das ist bei MPEG1 1152kBit/s. Die Option **-q** bestimmt den Kompressionsfaktor, der darüber entscheidet, wie genau die Bildinformation übernommen wird. Man sollte die Option so setzen, daß bei ruhigen Szenen weniger Bitrate gebraucht wird (etwa **-q 6** bis **-q 12**) und die maximale Bitrate mit der Option **-b** beschränken. Für Filme mit höchsten Qualitätsansprüchen (Archivierung) empfiehlt es sich, die Bitrate so hoch zu setzen, daß sie nie erreicht wird (so wären z. B. die Optionen **-b 10000** ( $10 \frac{MBit}{s}$ ) und **-q 2** oder **-q 3** eine gute Wahl).

Es folgt ein weiteres Beispiel für die Bildkomprimierung.

```
> lav2yuv Datei.avi | yuvscaler -I ACTIVE_352x240x+0+21 | mpeg2enc -b 1152 -r 16 -4 1 -2 1 -o Ausgabevideodatei.m1v
```

Viele Videofilme haben unten und oben einen fast schwarzen Rand. Dieser schwarzen Rand verbraucht auch Bandbreite, die man anders verwenden kann. Die Option **-I ACTIVE** des Programms yuvscaler setzt alles außerhalb des angegebenen Bereichs auf schwarz, aber die Bildgröße (352x288) wird dabei nicht verändert. Also gibt es dann schwarze Flächen, für die das Komprimierungsprogramm nur einige Bits benötigt. Man ist aber nach wie vor kompatibel zur VCD. Um das aktive Fenster zu bestimmen, extrahiert man aus dem Film ein Bild als Bild im jpeg-Format:

```
> lavtrans -f i -i 100 -o Bild.jpg Datei.avi
```

Das so extrahierte Bild wird mit einem Grafikprogramm bearbeitet, wobei man die schwarzen Ränder abschneidet. Es bleibt ein Bild mit der Größe des aktiven Fensters übrig. (Diesen Schritt kann man auch mit der CVS Version von LVS machen, wenn man nicht zur Kommandozeile greifen will). Die Optionen **-4 1** und **-2 1** verbessern die Bildqualität nochmals um etwa 10 %, verlangsamen aber die Komprimierung.

Bei einer Bildgröße von 352x288 (halbe PAL-Gri $\ddot{u}$  $\frac{1}{2}$ e, die man beim Aufnehmen mit der Option **-d 2** erzeugt) sollte man mit einer Bitrate zwischen 1 000 und 1 500  $\frac{kBit}{s}$  gute Ergebnisse erzielen.

Der wichtigste Faktor für Qualität ist aber die Qualität des Originals und wie man Filter einsetzt. Schlechtes ungefiltertes Material benötigt eine sehr hohe Bitrate, damit sichtbare

Artefakte vermieden werden. Um hier mit kleinerer Bitrate arbeiten zu können, ohne Artefakte beim Komprimieren von Fernsehfilmen zu bekommen, sollte man einen oder mehrere Rauschfilter einsetzen.

Weiters Beispiel:

```
> lav2yuv Datei.avi | mpeg2enc -b 1500 -n s -g 6 -G 20 -P -o Ausgabevideodatei.m1v
```

Hier wird das Rohmaterial mit einer Bitrate von  $1500 \frac{kBit}{s}$  komprimiert (Option **-b 1500**). Als Videonorm wird SECAM (**-n s**) erzwungen. Mit der Option **-P** wird sichergestellt, daß es immer 2 B Bilder zwischen angrenzenden I/P Bildern gibt. Viele verbreitete mpeg1-Decoder können Filme, bei denen das nicht der Fall ist, nicht abspielen. Mit den Optionen **-g 6 -G 20** wird dem Komprimierungsprogramm ermöglicht, die Anzahl der group-of-pictures zu ändern, um Szenenwechsel besser auszugleichen. Für VCDs sollte das Minimum nicht kleiner als 9 und das Maximum nicht größer als 15 sein. Bei SVCDs sollte man zwischen 6 und 18 bleiben. Wenn man sich allerdings auf die Wiedergabe mit Softwareplayern beschränkt, kann man auch andere Werte nehmen.

Nun folgt ein letztes Beispiel für dieses Kapitel:

```
> lav2yuv Datei*.avi | mpeg2enc -b 1500 -r 16 -4 1 -2 1 -S 630 -B 260 -o Ausgabevideo_n1_1500_41_21_S630_B260.m1v
```

Hier verarbeitet das Programm lav2yuv alle avi-Dateien, die ihren Namen mit Datei beginnen (Datei01.avi, Datei02.avi, ... und auch Datei\_keinvideo.avi, deshalb solle man eher mit %0xd Version arbeiten). Weiter hat das Programm mpeg2enc einige Optionen, die die Qualität des Films erhöhen, die alle schon weiter oben besprochen wurden. Die neue Option **-S 630** bedeutet, daß das Programm mpeg2enc nach einer Dateigröße von  $630 MB$  eine Marke setzt, so daß das Programm mplex an dieser Stelle eine neue Datei beginnt, wenn der Platzhalter %2d im Dateinamen angegeben ist. Wichtig, damit dies funktioniert, ist, daß man mit der Option **-B 260** die nicht-Video-Bitrate (Audio- und Multiplexinformation) angibt. Der Wert 260 sollte für eine Audiobitrate von  $224 \frac{kBit}{s}$  und die Multiplexinformation genügen. Mehr Information darüber gibt es in den encoding scripts im scripts Verzeichnis, wenn man die CVS oder tar source version verwendet.

### 9.5.3 Beispiele für das Zusammenführen der Dateien (multiplexen) im mpeg1-Format

Beispiel:

```
> mplex Ausgabetondatei.mp2 Ausgabevideodatei.m1v Mein_Video.mpg
```

Dieser Befehl fgt die Tondatei im mp2-Format und die Videodatei im mpeg1-Format zusammen zu dem Video Mein\_Video im mpg-Format. Dies funktioniert so einfach nur bei Filmen mit konstanter Bitrate (CBR). Das bedeutet, daß beim Komprimieren mit dem Programm mpeg2enc die Option **-q** nicht gesetzt war.

Ein Beispiel für einen Multiplexbefehl bei Filmen mit variabler Bitrate ist der folgende.

```
> mplex -V -r 1740 Ausgabetondatei.mp2 Ausgabevideodatei.m1v Mein_Video.mpg
```

Hier werden Dateien mit variabler Bitrate zu einem mpeg1-Film zusammengefügt. Da das mplex Programm den Film vor dem Zusammenfügen nicht lesen muß (single pass), kann es auch die maximale Bitrate nicht ermitteln. Daher muß man sie im Multiplexbefehl durch die Option **-r 1740** angeben. Die Bitrate des mpeg1-Films setzt sich zusammen aus **der Audiobitrate, dem Maximum der Videobitrate und zusätzlichen 1 bis 2 % für die Multiplexinformation**. Wenn die Audiodatei eine Bitrate von  $224 \frac{kBit}{s}$  hat (-b 224), das Video mit der Option **-b 1500 -q 9** komprimiert wurde, also eine Bitrate von  $1500 \frac{kBit}{s}$  hat, ergibt das eine Bitrate des fertigen Films von  $1724 \frac{kBit}{s} \cdot 1,01 = 1740 \frac{kBit}{s}$ .

## 9.6 Videofilme im mpeg2-Format erzeugen

Das Format mpeg2 sollte verwendet werden, wenn die Quellen Bilder mit mehr als 352x240 Pixel bei NTSC oder mehr als 352x288 bei PAL haben. Das Format mpeg2 kann auch mit Material umgehen, das interlaced ist (aus Halbbildern besteht), wie man es bekommt, wenn man aus dem Fernsehen mit voller Auflösung aufnimmt.

Das mpeg2-Format erlaubt die Verwendung des Formats mp3 für den Ton. Man kann zum Tonkomprimieren also seinen Lieblings-mp3-encoder verwenden. Auch das Audiosignal darf wie das Video- mit VBR (Variable BitRate) codiert sein.

Ein mpeg2-Film ist normalerweise ein VBR-Film, dessen Komprimierung mit Optimierungen sehr viel Rechenleistung verlangt. Deshalb dauert die Bildkompression eines Films auch je nach Leistungsfähigkeit der CPU sehr lange. Dabei hat aber ein Film mit der doppelten Auflösung **nicht** die vierfache Größe wie als mpeg1-Film. Je nach Qualitätseinstellung wird die Datei 1,5 bis 3 mal so groß sein wie als mpeg1-Film.

### 9.6.1 Beispiel für eine Audiodatei im mpeg2-Format

Beispiel:

```
> lav2wav Datei.avi | mp2enc -o Ausgabetondatei.mp2
```

Das funktioniert für das Format mpeg2 ohne Probleme. Man kann noch etwas Speicherplatz sparen, wenn man mit kleinerer Bitrate (Option **-b**) komprimiert. Brauchbar sind noch 160 oder  $192 \frac{kBit}{s}$ . Wenn der Ton 44.1kHz Sampling Rate haben soll muss man wieder **-r 44100** als Option zu dem Befehl hinzufügen. Und wie man einen mp3-encoder verwendet, muß wohl nicht erklärt werden. Man sollte dabei allerdings nicht alle besonderen Optionen verwenden, die der mp3-encoder bietet.

### 9.6.2 Beispiele für eine Videodatei im mpeg2-Format

Ein erstes Beispiel für die Komprimierung des Bildmaterials eines Films im mpeg2-Format ist der folgende Befehl.

```
> lav2yuv Datei.eli | mpeg2enc -f 3 -b 3000 -q 9 - Ausgabevideodatei.m2v
```

Dies ist ein einfaches Beispiel für die Erzeugung eines Videos im mpeg2 Format. Hierin ist die wichtigste Option die Option **-f 3**, mit der dem Programm mpeg2enc mitgeteilt wird, daß ein Film im mpeg2 Format erzeugt werden soll. Da es sich um einen allgemeinen mpeg2-file handelt, muß die Bitrate durch die Option **-b** mit angegeben werden. Es wird auch noch die Option **-q** verwendet, wodurch ein Film mit variabler Bitrate erzeugt wird, denn man will

ja eine möglichst kleine Dateigröße haben. Beim Erzeugen eines Films mit variabler Bitrate gibt die Option **-b** die maximale Bitrate an, die verwendet werden darf. Die Option **-q** teilt dem Programm mpeg2enc mit, welche Qualität der Film haben soll, soweit das im Rahmen der gewählten Bitrate möglich ist.

Ein zweites Beispiel für die Erzeugung eines Videos im mpeg2 Format ist der folgende Befehl.

```
> lav2yuv Datei.eli | mpeg2enc -f 3 -4 1 -2 1 -q 7 -b 4500 -V 300 -P -g 6 -G 18  
-I 1 -o Ausgabevideodatei.m2v
```

Das sieht schon mehr nach einem mpeg2 Film mit hoher Qualität aus. Die Optionen **-4 1** und **-2 1** machen etwas bessere Qualität. Durch die Optionen **-b 4500** und **-q 7** kennt das Programm mpeg2enc die gewünschte maximale Bitrate und den Qualitätsfaktor. Die Option **-V 300** gibt die Videopuffergröße in  $kB$  an, die für die Kompression verwendet werden soll. Wenn man noch Software playback hat, kann er wesentlich größer sein als die Voreinstellung. Die Option **-P** stellt sicher, das immer 2 B Bilder zwischen I/P Bildern sind. Viele mpeg-Decoder können nicht mit Filmen umgehen, bei denen weniger als 2 B Bilder zwischen den I/P Bildern sind. Mit der Option **-I 1** wird dem Programm mitgeteilt, daß es sich um interlaced Material handelt. Damit wird ein relativ zeitaufwendiges interlace-adapted motion compensation und block encoding Programm aufgerufen. Das Programm mpeg2enc schaltet diesen Mode automatisch ein, wenn die Bilder größer sind als VCD Größe bei der entsprechenden TV Norm.

Wenn man das Rauschfilterprogramm yuvdenoise zusätzlich verwendet, um das Bild zu verbessern, so kann man schon dort ein Deinterlacing mit der Option **-F** aktivieren. Dann sollte man natürlich dem Programm mpeg2enc mit der Option **-I 0** mitteilen, daß die Bilder schon deinterlaced sind. Außerdem muß dann keine zeitaufwendige motion estimation gemacht werden. Wird diese Option nicht verwendet, hat das keine anderen Effekte als höheren Zeitverbrauch.

Natürlich kann man auch hier skalieren, um die Bilder zu verbessern (denoise), damit man bessere Bilder bekommt und kleinere Dateien. Diese Dinge werden aber später beschrieben.

### 9.6.2.1 Werte der Option **-q** bei Filmen mit variabler Bitrate (VBR)

Die Option **-q** kontrolliert die minimale Quantisation des komprimierten Films. Die Quantisation kontrolliert die Genauigkeit, mit der die Bildinformation komprimiert wird. Je niedriger der Zahlenwert hinter der Option **-q**, desto besser wird die Bildqualität. Meistens muß noch zusätzlich eine maximale Bitrate mit der Option **-b** angegeben werden. Entscheidend ist nun, daß die Werte für die Optionen **-b** und **-q** so gewählt werden, daß man ein qualitativ gutes Video erhält, das nicht zuviel Bandbreite benötigt und auch noch nicht zu viele Artefakte enthält.

Der Qualitätsfaktor (**-q**) sollte so gewählt werden, daß man bei der Ausgabe des Multiplexen dann eine Differenz zwischen Spitze Peak-Bitrate und durchschnittlicher Bitrate von ca. 20 - 25 % hat. Wenn die Differenz sehr klein ist ( $< 10\%$ ), ist es sehr wahrscheinlich, daß man was falsch gemacht hat. Man hat wahrscheinlich eine zu niedrige Bitrate und/oder einen zu hohen Qualitätsfaktor gewählt. Die Zeilen sehen so aus:

```
INFO: [mplex] Average bit-rate : 2176800 bits/sec  
INFO: [mplex] Peak bit-rate : 6048400 bits/sec
```

Eine Kombination, die sicher mehr Artefakte produziert, als man zählen kann, ist, eine SVCD zu erstellen mit einer maximalen Bitrate von  $2500 \frac{kBit}{s}$  und einem Qualitätsfaktor



von 1 oder 2. Bei SVCDs mit einem Video Bandbreitenlimit von  $2\,500 \frac{kBit}{s}$  ist ein Qualitätsfaktor zwischen 7 und 12 etwa das Optimum. Dies gilt aber nur, wenn man Filter einsetzt, oder eine sehr gute Signalquelle wie Digital TV, DVD und ähnliches Material oder gerenderte Bilder hat. Da SVCD/DVD Spieler teilweise höhere Bitraten als die offiziellen  $2\,788 \frac{kBit}{s}$  für Audio und Video erlauben, kann man höhere Bitraten und einen höheren Qualitätsfaktor verwenden, damit auch Szenen mit sehr viel Bewegung sehr gut aussehen.

Bei einem Qualitätsfaktor zwischen 7 und 11 ist mit einer maximalen Bitrate von 3 500 bis 4 000  $\frac{kBit}{s}$  zu rechnen. Dann sollten auch nicht allzu viele Artefakte produziert werden. Für SVCD/DVD kann man bei ausreichend hoher maximaler Bitrate mit der im folgenden angegebenen Qualität des Films rechnen:

$q \leq 6$  sehr scharfe Bilder, sehr gute Qualität

$q \leq 8$  gute Qualität

$q \leq 10$  mittelmäßige Qualität

$q \leq 11$  nicht so gute Qualität

$q \leq 13$  Die Bilder könnten auch schon bei ruhigen Szenen Artefakte zeigen

### 9.6.2.2 Kompression für das Wiedergabeziel Fernsehen (interlaced) oder Monitor (progressive)

Das Format mpeg2 unterstützt sowohl interlaced Filme als auch progressive (not interlaced). Ein mpeg2 Video kann also interlaced oder progressive sein. Dies hängt ab von der Quelle (Film oder Fernsehen) und dem Wiedergabegerät.

Wird ein Film komprimiert, sollten zwei aufeinanderfolgende Felder eines Bildes immer gleich sein. Also kann man den Film mit dem Programm yuvdenoise **-F** deinterlacen, oder bei einer Quelle mit hoher Qualität, wo kein Rauschfilter benötigt wird, mit dem Programm yuvcorrect **-T NOT INTERLACED** deinterlacen. Bei dem Programm mpeg2enc sollte man dann die Option **-I 0** setzen. Damit wird dem Programm mitgeteilt, daß es kein Interlacing berücksichtigen muß. Deinterlacing wird nicht benötigt, da es keine Bewegung zwischen den beiden Feldern eines Bildes gibt. Man muß sie nur zu einem einzelnen progressiven Bild zusammenfügen.

Ein so erstellter Film sollte auf jedem Gerät (TV oder Monitor) problemlos abgespielt werden können.

Bei einer Aufnahmequelle mit interlaced Bildern (TV  $\ddot{u}$   $\frac{1}{2}$ ertragung) kann man den Film als interlaced komprimieren. Die Alternative ist, den Film zu deinterlacen und ihn als progressiven Film zu komprimieren. Aber durch das Deinterlacen mit dem Programm yuvdenoise **-F** gehen Details des Ausgangsmaterials verloren. Wenn man den Film auf dem DVD Abspielgerät und TV abspielen will, ist es nicht ratsam, das zu machen. Wenn man ihn aber nur auf seinem Monitor (progressives Display) abspielen will, wird das Bild besser aussehen, wenn es deinterlaced ist. Wenn der Player, den man verwendet, auch deinterlacing kann, ist es egal, ob das Video interlaced oder progressive Bilder hat.

Wenn man für das Deinterlacing das Programm yuvdenoise **-F** benutzt, sollte man nicht vergessen, bei dem Programm mpeg2enc die Option **-I 0** zu setzen. Wenn man den Film nicht deinterlacen will, benötigt man keine besonderen Optionen (weder bei yuvdenoise die Option **-F** noch bei mpeg2enc die Option **-I 0**).

Wenn man gerne Filme stoppt, um Standbilder anzusehen, ist es besser, den Film zu deinterlacen, weil das Standbild dann flimmerfrei ist.

Ist ein progressiver Film mit TV Anteilen gemischt, wird es problematisch. (Das gibt es z.

B. bei Dokumentationen. Der Sprecher wurde mit einer Videokamera aufgenommen (interlaced) und die Filmsequenzen sind progressive.) In dem Fall muß man entscheiden, ob man optimale Filmsequenzen mit guten Standbildern haben will, dann sollte man deinterlacen (yuvdenoise **-F** mpeg2enc **-I 0**). Andernfalls muß man nicht deinterlacen, wodurch man dann einer bessere Qualität bei den interlaced Teilen hat.

### 9.6.3 Beispiel für das Zusammenführen der Dateien (multiplexen) im mpeg2-Format

Das Zusammenführen der komprimierten Audio- und Videodateien kann z. B. mit folgendem Befehl erfolgen:

```
> mplex -f 3 -b 300 -r 4750 -V Ausgabedatei.mp3 Ausgabevideodatei.m2v  
MeinVideo.mpg
```

Die Audio- und die Videodatei werden zu einem Film zusammengeführt. Als Quelldateien werden eine Audiodatei im Format mp3 und eine Videodatei im mpeg2 Format verwendet. Die Option **-f 3** teilt dem Programm mplex mit, daß als Ausgabeformat eine mpeg2-Datei gewünscht wird. Man muß durch die Option **-b 300** die gleiche Pufferspeichergröße angeben, die auch beim Komprimieren verwendet wurde. Die Option **-r 4750** gibt die maximale Bitrate für die Audio-, die Videodatei und für die Multiplexinformation an. Mit der Option **-V** wird der Film als Film mit variabler Bitrate (VBR) zusammengeführt. Wenn man diese Option nicht verwendet, wird ein Film mit konstanter Bitrate (CBR) erzeugt. Dann ist der Film so groß wie ein Film mit konstanter Bitrate ist, der mit der Datenrate erzeugt hat, die in der Option **-r** angegeben ist. Diese Filme mit konstanter Bitrate sind normalerweise sehr groß.

## 9.7 Video-CD (VCD) erzeugen

Das VCD-Format ist eine eingeschränkte Version des mpeg1-Formates, das von der Firma Philips entwickelt wurde. Die Spezifikation wurde von Philips nicht freigegeben, daher ist sie nicht frei zugänglich. Das Ziel bei der Entwicklung dieses Formates war es, ein billiges Abspielgerät aus einem CD-Rom Laufwerk mit einfacher Geschwindigkeit zu bauen. Die Videobitrate für dieses Format beträgt  $1152\text{ kBit/s}$ , die Audiobitrate im Layer 2-Format  $224\text{ kBit/s}$ . Für dieses Format darf beim Programm mpeg2enc die Option **-q** nicht verwendet werden. Weiters ist die Verwendung von dynamischen GOPs eingeschränkt und der Videopuffer ist auf  $46\text{ kB}$  beschränkt.

Wenn man keinen VCD Spieler hat, kann man die VCD-Filme auf einem DVD Spieler abspielen. Hierbei kann es sein, daß der DVD Spieler nicht so restriktiv ist und größere Bitraten, dynamische GOPs, größeren Videopuffer variable Bitrate und so weiter zuläßt.

### 9.7.1 Beispiel für die Tonkomprimierung für eine VCD

Die Tonkomprimierung kann z. B. mit dem Befehl

```
> lav2wav Datei.avi | mp2enc -V -o Tondatei.mp2
```

Die Option **-V** erzwingt ein VCD 2.0 kompatibles Ausgabeformat Die Smpelrate ist mir  $44.1\text{ kHz}$  vorgegeben. Man kann bei Mono Ton zwischen  $64$ ,  $96$  oder  $192\text{ kBit/s}$  wählen und

bei Stereo zwischen 128, 192, 224 oder 384kBit wÄ?hlen. Wenn der Film auf einen separaten Abspielgerät wiedergegeben werden soll, sollte man für das Audioformat nur das Layer 2-Format (mp2) mit 44,1 kHz Stereo 224 kBit/s verwenden.

### 9.7.2 Beispiel für die Videokomprimierung fr eine VCD

Ein Beispiel für eine Bildkompression für eine VCD ist der Befehl

```
> lav2yuv Datei.avi | yuvscaler -0 VCD | mpeg2enc -f 1 -r 16 -o Videodatei.m1v
```

Die Option -f 1 sorgt dafür, daß das Programm mpeg2enc einen VCD kompatiblen Film erzeugt. Für eine VCD sollten **nie** die Optionen -b, -m oder -q verwendet werden. Anscheinend haben manche DVD Abspielgeräte (z. B. Avex) Probleme, wenn die mpeg-Filme mit einem Suchradius größer als 16 komprimiert wurden. Die Voreinstellung -r 16 sollte also nie durch einen größeren Wert ersetzt werden.

Ein weiters Beispiel ist der folgende Befehl.

```
> lav2yuv Datei.avi | mpeg2enc -f 1 -4 1 -2 1 -S 630 -B 260 -P -o Videodatei.m1v
```

Die Option -S 630 setzt beim Komprimieren eine Marke, so daß das Programm mplex beim Zusammenfügen von Bild und Ton nach einer Dateigröße von 630 MB eine neue Datei anfängt. Damit das auch funktioniert, muß zusätzlich zur Option -S 630 die Audio- und Multiplexbitrate mit der Option -B angegeben werden. Ein Wert von -B 260 sollte für die Audio- und die Multiplexbitrate genügen. Mehr Information dazu findet sich bei den encoding Skripts im Skripts Verzeichnis. Der Film mit Ton- und Bildspur sollte so leicht auf eine CD mit 650 MB Speicherplatz passen.

Die Voreinstellung für die Option -S ist 700 MB. Das Programm mpeg2enc markiert automatisch jeden Film bei der Größe von 700 MB, wenn die Option -S nicht etwas anderes angibt. Wenn man einen CD-Rohling hat, auf den z. B. 800 MB passen, sollte die Option -S gesetzt werden. Andernfalls wird das Programm mpeg2enc den Film bei einer Größe von 700 MB markieren. Dadurch wird das Programm mplex den Film so teilen, daß nach einer Dateigröße von 700 MB eine neue Datei angefangen wird. Das ist bei einem 800 MB Rohling sicher nicht das, was man will. Den im XA Mode2 passen mehr Daten in jeden Sektor der CD. Es sind 2324Byte statt 2048. Deshalb passen im Mode 2 etwas mehr als 700 MB auf einen Rohling der als 650 MB Rohling verkauft wird.

### 9.7.3 Beispiel für das Zusammenführen von Ton und Bild für eine VCD

Der Befehl dazu lautet

```
> mplex -f 1 Tondatei.mp2 Videodatei.m1v -o VCDdatei.mpg
```

Die Option -f 1 erzeugt viel Information, die normalerweise nicht benötigt wird, aber für die Erstellung einer VCD notwendig ist.

## 9.7.4 Erzeugung der VCD

Der zusammengeführte Film muß nun mit Informationen ergänzt werden, damit das Abspielgerät bestimmte Daten (Zeit, Navigationsinformation) bekommt. Die kann das Programm `vcdimager` erstellen, das man von der Adresse <http://www.vcdimager.org/> bekommt. Der Programmaufruf lautet

```
> vcdimager testvideo.mpg
```

Hierbei werden ein Datenfile mit dem Namen `videocd.bin` und ein Kontrollfile mit Namen `videocd.cue` für das Brennprogramm `cdrdao` erzeugt. Unter Linux verwendet man `cdrdao` zum Brennen des Images. Dieses Brennprogramm findet sich bei <http://cdrdao.sourceforge.net/index.html>. Das eigentliche Brennen kann dann mit einem GUI wie `gcdmaster` erfolgen, z. B. durch den Aufruf

```
> gcdmaster videocd.cue
```

Für die Komprimierung einer Show (45 Minuten) oder eines 90 minütigen Filmes vom Fernsehen dürfte eine Bitrate von  $1\,800\text{ kBit/s}$  ideal sein. Dies ergibt eine Datei mit etwa  $700\text{ MB}$  Größe und  $45\text{ min}$  Laufzeit, die gut als raw XA-MODE2 Datentrack auf eine CD passen. Bei digitalen Quellen (DTV oder DVD und ähnlichen) genügt für eine VCD schon eine Bitrate von  $1\,152\text{ kBit/s}$ .

**Hinweis:** Wenn man mit variabler Bitrate (VBR, Option `-q`) arbeitet, kann es sein, daß das Abspielgerät die CD nicht abspielen kann, da er dafür nicht spezifiziert wurde. Wenn es doch funktioniert, sollte man sich einfach freuen. Manchmal hilft es auch, wenn man mit dem Programm `vcdimager` den Film als SVCD erzeugt. Das Programm `vcdimager` beschwert sich dann zwar, aber nur mit einer Warnung und nicht mit einer Fehlermeldung. Man sollte den so erzeugten Film dann doch brennen können. Dies kann einige Abspielgeräte überzeugen, den Film jetzt abzuspielen; dafür gibt es aber keine Garantie.

## 9.7.5 MPEG-Filme speichern

Wenn man die Daten als XA-MODE2 Track speichert, paßt mehr auf eine CD, aber es gibt dann keine Fehlererkennung und Fehlerkorrektur mehr. Man kann das Programm `vcdimager` nutzen um dies zu tun. Mit dem Programm `vcdxrip` des Programmpaketes `vcdimager` kann man von bereits gebrannten VCD die Filme wieder extrahieren. Man erhält wieder MPEG videos und ein XML mit der Struktur der VCD. Bessere Qualität erhält man mit den Formaten SVCD, XVCD und DVD.

Wenn das Abspielgerät das SVCD-Format nicht unterstützt, kann es möglicherweise Filme im VCD-Format mit einer wesentlich höheren Bitrate als der Standard abspielen. Meist können Filme mit Bitraten bis zu  $2\,500\text{ kBit/s}$  verarbeitet werden. Bei diesen Bitraten und qualitativ gutem Ausgangsmaterial lohnt es sich, die Option `-h` beim Komprimieren mit `mpeg2enc` zu probieren. Der Film sollte damit so scharf werden, wie es die Grenzen des VCD-Formates erlauben. Die Option `-h` hilft auch bei Ausgangsmaterial von schlechter Qualität. Hier wird der Film zwar nicht so scharf, dafür bleibt das Bild ruhig.

Wenn das Abspielgerät das SVCD-Format unterstützt und man bereit ist, die dafür längere Kompressionszeit in Kauf zu nehmen, ist dieses Format sicher die bessere Wahl. Durch das effizientere `mpeg2`-Format hat man eine mehr als doppelt so große Auflösung im Vergleich

zum mpeg1-Format, wobei die Dateien aber weit unter dem Faktor 2 größer sind als die mpeg1 Dateien.

## 9.8 Super Video CD (SVCD) erzeugen

Das SVCD-Format ist eine verbesserte Version des VCD-Formates. Es wurde von einem von der chinesischen Regierung ins Leben gerufenen Komitee aus Herstellern und Entwicklern entwickelt. Die endgültige Spezifikation wurde 1998 festgelegt. Eine gute Erklärung des SVCD-Formates von Philips kann unter

<http://licensing.philips.com/information/cd/video/documents575.html>  
eingesehen werden.

Die Aufnahme muß für dieses Format mit voller TV Auflösung (also -d 1, das ist bei PAL 720x576 oder NTSC 720x480) erfolgen. Denn die SVCD Auflösung ist bei NTSC 480x480 und bei PAL 480x576, diese Auflösung ist wesentlich höher als die halbe Breite. Da man nicht aufgenommen Information schlecht errechnen, sprich hinaufskalieren, kann, sollte man, um möglichst viele Details zu erhalten, in voller Breite aufnehmen, und dann herunterskalieren auf eine Breite von 480 Pixel. Prinzipiell funktioniert es auch von halber Breite, also 352 Pixel hinaufzuskalieren, aber dann muß man damit rechnen, daß man weniger Details hat.

### 9.8.1 Beispiel für die Tonkomprimierung für eine SVCD

Die Tonkomprimierung kann z. B. mit dem Befehl

```
> lav2wav Datei.avi | mp2enc -V -e -o Tondatei.mp2
```

erfolgen. Hierbei erlaubt die SVCD Spezifikation eine breitere Wahl der Audiobitrate. Erlaubt sind Bitraten zwischen 32 und 384 *kBit/s*, wobei auch für die Audiobitrate variable Bitraten (VBR) erlaubt sind. Die Option -e schaltet den CRC (Cylindric Redundancy Check) Fehlerschutz des Tons an, der einschaltet sein muß um SVCD konform zu sein. Aber es scheint, daß die meisten Abspielgeräte auf die CRC Information gar nicht achten. Die CRC Information benötigt 2 Bytes pro Audioframe.

Die zu erwartende Framegröße je Audioframe bei mpeg1 Layer 2 berechnet sich wie folgt:

$$\text{Signalgröße} = 144 \cdot \frac{\text{Bitrate}}{\text{Samplerate}} \text{ Bytes} = 144 \cdot \frac{224\text{kBit/s} * 1024 \frac{\text{Bit}}{\text{kBit}}}{44100 \frac{\text{Bit}}{\text{s}}} = 749 \text{ Byte}$$

Beim der typischen VCD Einstellung (224kBit, 44.1kHz) benötigt die CRC Information etwa 0,27 % der Gesamtsignalgröße, also von 815 auf 817 Bytes. Im ungünstigsten Falle eines Tons mit Mono 32 *kBit/s* Bitrate benötigt die CRC Information etwa 1,92 % der Gesamtdaten.

### 9.8.2 Beispiel für die Videokomprimierung für eine SVCD

Die Bildkompression kann für eine SVCD z. B. mit folgendem Befehl vorgenommen werden:

```
> lav2yuv Datei.avi | yuvscaler -0 SVCD | mpeg2enc -f 4 -q 7 -I 1 -V 200  
-o Videodatei.m2v
```

Die Option

-f 4

teilt dem Programm mpeg2enc mit, daß es für eine SVCD formatieren soll.

Mit

- q 7

wird eine variable Bitrate mit der Qualitätsstufe 7 erzeugt.

Die Option

-I 1

informiert das Programm mpeg2enc darüber, daß es sich bei dem Film um interlaced Material handelt, wo die ungeraden Pixelreihen ein Halbbild bilden und die geraden Pixelreihen das folgende zweite Halbbild. Die Option -I 0 (progressives Ausgangssignal, keine Halbbilder) funktioniert aber bei PAL Signalen auch.

Man kann kleinere Bitraten verwenden. Es ist aber zu beachten, daß das Format SVCD die **totale Bitrate (Audio und Video) 2 788 800  $\frac{\text{Bit}}{\text{s}}$**  limitiert. Eine Audiobitrate von  $224\text{kBit/s}$  zusammen mit einer Videobitrate von  $2\,550\text{kBit/s}$  sowie der notwendigen Kopf-information kann also knapp zuviel sein. Da aber das SVCD-Format Audiobitraten zwischen  $32$  und  $224\text{kBit/s}$  erlaubt, kann man einige Bits sparen, indem man für das Audiosignal eine Bitrate von  $192\text{kBit/s}$  (oder  $160\text{kBit/s}$  für nicht Musikmaterial) verwendet.

Das SVCD-Format unterstützt variable Bitraten (VBR), da mpeg2 normalerweise mit variabler Bitrate vorliegt, aber mit einer maximalen Videobitrate von  $2\,500\text{kBit/s}$ . Mit der Option -f 4 setzt das Programm mpeg2enc auch gleich dynamisches GOP mit einer unteren Grenze von -g 6 und einer oberen Grenze von -G 18. Dies spart einige Bits ein und verbessert die Bildqualität bei Szenenwechseln. Wenn mit der Option -f 4 komprimiert wird, ignoriert das Programm mpeg2enc die Optionen -b (Videobitrate) und -r (Suchradius). Wird statt -f 4 die Option -f 5 gesetzt, müssen aber die Videobitrate und die anderen Optionen für mpeg2enc gesetzt werden.

Ein weiterer möglicher Kompressionsbefehl für PAL-Signale mit  $25\text{ Bilder/s}$  ( $50\text{ Halbbilder/s}$ ) ist folgender Befehl:

```
> lav2yuv datei.avi | yuvscaler -0 SVCD | mpeg2enc -f 4 -I 0 -V 300  
-o Videodatei.m2v
```

Kinofilme werden üblicherweise mit  $24\text{ Bildern/s}$  aufgenommen. Im PAL Fernsehen werden diese Filme normalerweise dann einfach etwas zu schnell abgespielt ( $25\text{ Bilder/s}$ ), was von Personen mit einem guten Geschwindigkeitsgefühl wahrgenommen und als störend empfunden wird. Die Option -I 0 unterbindet die langwierigen Berechnungen für die Kompensation des Interlacing, wodurch das Komprimieren schneller wird.

Unglücklicherweise lassen sich NTSC-Übertragungen (US Format mit  $30\text{ Bilder/s}$  ( $60\text{ Halbbilder/s}$ )) nur sehr schlecht komprimieren, wenn sie von Filmen mit  $24\text{ Bilder/s}$  erzeugt wurden. Das verwendete „pulldownsampling“, um die  $60$  Halbbilder zu erzeugen aus einem  $24$  Frame Film bedeutet, daß die Hälfte der Frames field interlaced sind. Mehr Information dazu gibt es in der manpage von yuvkineco.

Im Kompressionsbefehl sollten zusätzlich die Optionen -S und -B gesetzt werden, die weiter oben beschrieben wurden, wenn der Film auf eine CD passen soll.

### 9.8.3 Beispiel für das Zusammenführen von Ton und Bild für eine SVCD

Der Multiplex-Befehl lautet für eine SVCD

```
> mplex -f 4 -b 300 -r 2750 Tondatei.mp2 Videodatei.m2v -o SVCD.mpg
```

Die Option

-f 4

informiert das Multiplexprogramm, daß es eine SVCD generieren soll.

Die Option

-r 2750

ist die kalkulierte Audio und Video Bitrate zuzüglich 1 - 2 % für die Multiplexinformation. Schließlich gibt

-b 300

die Puffergröße an, die für das Abspielgerät zur Verfügung steht. Es ist derselbe Wert, der für die Kompression verwendet wurde (Option -V beim Programm mpeg2enc).

### 9.8.4 Erzeugung der SVCD

Die Erzeugung der SVCD erfolgt wieder mit dem Programm vcdimager. Der Befehl dazu lautet

```
> vcdimager -t svcd SVCD.mpg
```

Dies erzeugt wieder die beiden Dateien **videocd.bin** und **videocd.cue**, die dann mit dem Befehl **gcdmaster videocd.cue** gebrannt werden können.

Hierzu noch zwei Hinweise:

- Wenn normale VCD/SVCD erzeugt werden sollen, kann man auch die Optionen -f 2 und -f 5 beim Programm mplex versuchen.
- Ob die so erzeugten CDs auf allen Abspielgeräten funktionieren, kann nicht garantiert werden. Es gibt viele Berichte, daß es recht gut funktioniert. Ärgern Sie sich nicht, wenn es nicht funktioniert. Weiter bin ich nicht verantwortlich für nicht brauchbare und verbrannte CDs.

## 9.9 Erzeugung einer DVD

Die Beschränkungen, die im folgenden erwähnt sind, können in der gerade aktuellen Version schon aufgehoben sein.

Zum Erstellen einer DVD brauchen Sie natürlich einen DVD-Brenner, die aber heute schon zu erschwinglichen Preisen mit 8-facher Brenngeschwindigkeit im Markt erhältlich sind. Zum Erzeugen der DVD und den dort notwendigen Dateien VOB, IFO und BUP wird das Programm dvdauthor benutzt, das unter

<http://dvdauthor.sourceforge.net>

erhältlich ist.

### 9.9.1 Beispiel für die Tonkomprimierung für eine DVD

Der Befehl hierzu lautet

```
> lav2wav Datei.eli | mp2enc -r 48000 -o Tondatei.mp2
```

Die Option

```
-r 48000
```

setzt die Samplerate auf  $48\,000\text{ Hz} = 48\text{ kHz}$ , was zwingend erforderlich ist. Mit dieser Option wird das Programm `mp2enc` gezwungen, diese Samplerate zu verwenden. Wenn die Sempelrate nicht schon  $48\text{ kHz}$  ist, wird das Programm `mp2enc` resampeln, um auf diese Sempelrate zu kommen. Beträgt die Sempelrate schon  $48\text{ kHz}$ , kann zum Komprimieren des Tons statt des Programms `mp2enc` auch das Programm `toolame` verwendet werden.

### 9.9.2 Beispiel für die Videokomprimierung für eine DVD

Die Bildkomprimierung erfolgt z. B. mit dem Befehl

```
> lav2yuv Datei.eli | mpeg2enc -f 8 -o Videodatei.m2v
```

Mit der Option

```
-f 8
```

erzeugt das Programm `mpeg2enc` ein `mpeg2`-Video, das sehr nahe am DVD Standard ist. Die Bitrate wird auf  $7\,500\text{ kBit/s}$  gesetzt und der Videopuffer auf den bei DVDs maximal zulässigen Wert. Der Qualitätsfaktor wird auf 8 gesetzt. Das Programm `mpeg2enc` setzt derzeit keine automatische Sequenzlänge, wie es das bei der Erzeugung von VCD und SVCD macht.

Alle Optionen, die auch sonst beim Programm `mpeg2enc` verwendet werden können, sind auch hier erlaubt. Ebenso kann auch hier die Bildqualität mit den Programmen `yuvscaler` und `yuvdenoise` verbessert werden. Ein vollst2andiges Kommando zum Erzeugen eines DVD nahen `mpeg2` Videos könnte dann z. B. so aussehen:

```
> lav2yuv Datei.eli | yuvdenoise -F -f | mpeg2enc -f 8 -q 7 -4 1 -2 1 -P -I 0  
-o Videodatei.m2v
```

### 9.9.3 Beispiel für das Zusammenführen von Ton und Bild für eine DVD

Der Multiplexbefehl sieht z. B. so aus:

```
> mplex -f 8 Tondatei.mp2 Videodatei.m2v -0 DVDij1/2nlich.mpg
```

Mit der Option



-f 8

wird dem Programm `mplex` mitgeteilt, daß es ein `mpeg2`-Video erzeugen soll, das sehr nahe am DVD Standard ist. Das Programm `mplex` kann nicht alle Dinge, die der DVD Standard erlaubt. Das Ergebnis, das `mplex` erzeugt, liegt aber so nahe beim DVD Standard, daß die DVD Abspielgeräte es akzeptieren.

### 9.9.4 Erzeugung der DVD

Für die Erstellung der eigentlichen DVD wird das Programm `dvdauthor` benutzt, das unter <http://dvdauthor.sourceforge.net> erhältlich ist. Die Bedienung ist in der zugehörigen Dokumentation sehr ausführlich erläutert, so daß hier nur eingeschränkt darauf eingegangen wird. Im Zweifelsfall sollte man in der Originaldokumentation nachlesen. Der grundsätzliche Befehl hierzu sieht etwa so aus:

```
> makedvd stream1.mpg stream2.mpg ... DVDVideo.mpg
```

Es wird ein `AUDIO_TS` und ein `VIDEO_TS` Verzeichnis angelegt. Wo das Verzeichnis angelegt wird, hängt von der Einstellung des `BASEDIR` im `makedvd` Script ab. Diesem Verzeichnis wird ein `AUDIO_TS` und `VIDEO_TS` Verzeichnis angelegt, von wo man dann die DVD brennen kann. Bei einem DVD +R/+RW-Brenner sieht der Brennbefehl dann etwa so aus:

```
> growisofs -Z /dev/scd2 -dvd-video mydvd/
```

Genauere Erklärung zu dem Befehl oder einen alternativ zu verwendenden Befehl findet man in der Dokumentation zu `dvdauthor` und dem für das DVD Format passenden Brennerprogramm.

## 9.10 DIVX-Videos erzeugen

DIVX-Videos können mit dem Programm `mencoder` erzeugt werden. Dieses Programm ist ein Teil des Programmpaketes `MPlayer`, das unter <http://www.mplayer.hu> bezogen werden kann.

Die Ausgabedaten des Programmes `mencoder` werden von dem `mjpegtools` shell-script `lav2avi.sh` als Eingangsinformation benutzt. Der erste und zweite Durchgang von `mencoder` liefern am Ende Hinweise über die Bitrate. Diese Information kann verwendet werden, um auf eine bestimmte Größe zu komprimieren (650 MB, 700 MB und 800 MB). Das weitgehend selbsterklärende Skript `lav2avi.sh` verwendet diese Informationen, wenn sie vom Programm `mencoder` zur Verfügung gestellt wird, was bei zu kurzen Filmen nicht der Fall ist. Der Parameter `preferredSize` gibt die Endgröße an. Man kann natürlich auch weitere Parameter für die Codierung verwenden, wenn man im Script die Option `encoderParam` verwendet. Eine Beschreibung der möglichen Optionen steht in der Dokumentation des Programmpaketes `MPlayer`.

Die Ausgabedatei hat den gleichen Namen wie die Eingangsdatei, nur wird die Endung gegen die Endung `.avi` getauscht. Sollte die Datei kleiner sein als mit der Option `preferredSize` gewünscht, so kann das daran liegen, daß es für die Längenberechnung zu wenig Informationen gab. Üblich ist, daß ein 1,5 h dauernder Film bei halber Bildgröße ein etwa 700 MB großes Video ergibt. Das wäre eine Bitrate von etwa  $900 \frac{kBit}{s}$ , was für ein DIVX-Video schon eine recht gute Qualität bedeutet.

Das Script codiert in drei Schritten:

1. Schritt: Audio codieren
2. Schritt: Video codieren erster Durchgang
3. Schritt: Video codieren zweiter Durchgang.

Für die Komprimierung wird der **fast ffmpeg codec** (compilation switch von mencoder) verwendet. Das ergibt recht gute Ergebnisse bei relativ hoher Kompressionsgeschwindigkeit. Für die Audiokomprimierung wird mp3 als Codec verwendet. Genauere Information über die Codierung findet man in der Dokumentation von MPlayer/mencoder.

Wenn man das shellscript den eigenen Bedürfnissen anpassen will, sollte man sich folgender Punkte bewußt sein:

- Die Ausgabedatei des ersten Schrittes wird **frameno.avi** genannt und enthält nur die Audioinformation.
- Im zweiten Schritt wird diese Ausgabedatei des ersten Schrittes verwendet und eine Textdatei mit dem Namen **lavc\_stats.txt** erzeugt. Diese Datei enthält die Timing Informationen.
- Im dritten Schritt werden die Dateien **frameno.avi** und **lavc\_stats.txt** verwendet, um die Ausgabedatei **movie2.avi** zu erzeugen.
- Wenn man die Videobitrate ändern will, darf die Datei **frameno.avi** nicht gelöscht werden. Dazu muß man im Skript das Löschen mit **#** auskommentieren. Man sollte dann den ersten Schritt nicht mehr ausführen und einfach nur den zweiten und dritten Schritt wiederholen.

# 10 Filmoptimierung

## 10.1 Einsatz von Filterprogrammen

Die Qualität der erzeugten Filme wird bei Verwendung von konstanter Bitrate (CBR) durch den Einsatz von Filterprogrammen erhöht. Bei variabler Bitrate (VBR) benötigt der codierte Film weniger Speicherplatz als ohne Filter. Ein Beispiel für den Einsatz eines Filters zur Qualitätsverbesserung bei konstanter Bitrate ist der folgende Befehl.

```
> lav2yuv Datei.avi | yuvmedianfilter | mpeg2enc -o video.m1v
```

Hier wird das Programm `yuvmedianfilter` benutzt um die Bildqualität zu verbessern. Das Programm entfernt niederfrequentes Rauschen, außerdem werden die Bilder etwas weichgezeichnet. Das Programm nimmt dazu einen Bildpunkt und summiert die um diesen Punkt herum liegenden Pixel, die unter die Rauschschwelle fallen. Diese Pixel werden dann mit einem neuen Wert überschrieben. Als Optionen kann man dem Programm `yuvmedianfilter` den Suchradius (`-r`), um den gesucht werden soll, sowie die Rauschschwelle (`-t`) mitgeben, über die der Pixel gemittelt wird. Wird der Suchradius größer als 2, benötigt das Programm sehr viel Zeit.

```
> lav2yuv Datei.avi | yuvmedianfilter -r 3 -t 4 -T 0 | mpeg2enc -o video.m1v
```

In diesem Beispiel wird ein anderer Suchradius als der Wert der Voreinstellung (default ist 2) von 3 Pixel für luma (Helligkeit) und eine Rauschschwelle von 4 (`-t 4` (threshold)) (Voreinstellung ist 2) verwendet. Außerdem wird die Filterung von chroma (Farbe) durch die Option `-T 0` ausgeschaltet. Abhängig von der Aufnahmequelle kann es passieren, daß eine chroma-Filterung eine leichte Farbverschiebung nach grün bewirkt. Dann hilft es, wie im Beispiel gezeigt den Filter für chroma zu deaktivieren und nur luma zu filtern.

```
> lav2yuv Datei.avi | yuvdenoise | mpeg2enc - video.m1v
```

In diesem Beispiel wird das Programm `yuvdenoise` verwendet, um die Bildqualität  $\frac{1}{2}$  zu verbessern. Dieser Filter verringert neben Farb- und Helligkeitsrauschen auch Flackern aufgrund von Phasenfehlern. Soll dieses Filterprogramm den Film deinterlacen, muß die Option `-F` gesetzt werden.

Voreinstellungsgemäß wird das Programm `yuvdenoise` den Film als interlaced Material filtern. Wie beim Filterprogramm `yuvmedianfilter` kann man auch hier durch die Optionen `-t` und `-r` die Rauschschwelle und den Suchradius verändern. Es kann auch eine **active area** gesetzt werden, bei der dann alles außerhalb dieser Fläche auf schwarz gesetzt wird. Wie man eine **active area** setzt, ist in der manpage von `yuvdenoise` beschrieben. Dies ist vor allem bei VHS und 8 mm-Videoquellen sinnvoll, denn diese haben immer am unteren Rand einige Zeilen mit hohem Rauschen. Neben `yuvdenoise` können auch noch `yuvscaler` und `y4mshift`

eine **active area** setzten. Das Programm yuvinactive macht genau das Gegenteil, es setzt die angegebene Fläche auf schwarz oder behandelt die Fläche wie in den Optionen angegeben. Das Programm yuvdenoise verwendet einen anderen Ansatz, um das Bild zu filtern, als das Programm yuvmedianfilter. Eine genaue Beschreibung sowie alle weiteren möglichen Optionen finden sich in der manpage des Programms.

Wenn das Quellmaterial von sehr guter Qualität ist, kann man auch den **fast mode -f** verwenden, bei dem das Programm yuvdenoise lediglich eine Reduktion des Bitrauschens macht. In diesem Modus kann die Option **-h/-keep-hf** des Programms mpeg2enc auch hilfreich sein, denn mit dieser Option wird das Programm mpeg2enc angewiesen, möglichst viel Bildinformation mit hoher Frequenz zu speichern. Diese Option vergrößert aber die Bitrate beträchtlich. Sollte die Bitrate zu nahe am Maximum (Option -b) liegen, wird das Programm die Qualität wieder zurücknehmen, damit die maximale Bitrate nicht überschritten wird.

Eine weitere Möglichkeit der Qualitätsverbesserung ist die Option **-N/-reduce-hf** des Programms mpeg2enc, die gleichermaßen einen eingebauten Filter darstellt. Mit dieser Option wird die Genauigkeit, mit der die Bildinformation mit hoher Frequenz codiert wird, geändert. Hohe Frequenzen treten z. B. an Kanten oder bei Farbwechsel auf. Die Option -N kontrolliert die Genauigkeit, mit der diese Information gespeichert wird. Erlaubt sind Werte zwischen 0.0 und 2.0, wobei der Wert 0.0 den Filter (frequency quantiser boost) deaktiviert und 2.0 das Maximum darstellt. Welchen Wert man verwendet, hängt davon ab, was man erreichen will. Bei Werten unter 0.5 sind die Veränderungen minimal, während Werte um 1.0 einen guten Kompromiß darstellen zwischen kleiner Bitrate und mehr Bildveränderung. Bei Werten über 1.5 gibt es einen sichtbaren Schärfeverlust im Bild. So große Werte sollte man deshalb nur bei Quellen verwenden, die sowieso kein besonders scharfes Bild liefern wie z.B. VHS.

Die Kombination der Programme yuvdenoise, yuvmedianfilter und der mpeg2enc-Option -N liefert eine gute Kontrolle über die Bitrate und damit der Dateigröße. Die Verkleinerung (und Vergrößerung) der Bitrate hängt von der Videoquelle und den verwendeten Optionen ab. Es ist leider nicht möglich exakt anzugeben, welche Kombination von Optionen welches Ergebnis bringt, sondern nur Richtlinien.

Als Richtlinie kann man folgende Hinweise nehmen: Die Option -N des Programms mpeg2enc sollte in einem Bereich zwischen 0.5 und 1.5 verwendet werden. Werte unter 0.5 bringen nicht viel (aber die Schärfe bleibt), bei Werten über 1.5 wird das Ergebnis schon recht weichgezeichnete Bilder haben. Es ist auch möglich, daß es bei scharf begrenzten Objekten (Text) bei Werten über 1.5 zu Schattenbildung (halo/ringing) kommt. Wenn man die Filter verwendet, sollte man das Programm yuvmedianfilter in der Befehlskette nach dem Programm yuvdenoise verwenden. Weiter sollte das Programm yuvmedianfilter nach dem Programm yuvscaler einsetzen. Dabei wird bei Ausgangsmaterial von guter Qualität die Bitrate durch die zusätzliche Verwendung des Programms yuvmedianfilter im Gegensatz zu Ausgangsmaterial von schlechter Qualität nicht wesentlich gesenkt. Grob kann man sagen, daß durch den Einsatz der Filter die benötigte Bitrate gegenüber ungefiltertem Material etwa halbiert wird.

Allgemein kann gesagt werden, daß mit zunehmender Filterung die Dateien immer kleiner werden (kleinere Bitrate), daß aber im Gegenzug Details vom Bild immer mehr verschwinden, wohingegen weniger aggressives Filtern größere Dateien mit mehr Details ergibt.

Es folgt ein weiteres Beispiel speziell zum Verbessern von NTSC Quellmaterial.

```
> lav2yuv Datei.avi | yuvkineco -F 1 | mpeg2enc -o video.m1v
```

Das Programm yuvkineco ist nur für die Verbesserung von NTSC-Quellmaterial, das die Konversion von  $\frac{30000.0}{1001.0} \approx 29,97 \frac{\text{Bilder}}{\text{s}}$  auf  $\frac{24000.0}{1001.0} = 23,976 \frac{\text{Bilder}}{\text{s}}$  durchführt. Das Programm

yuvkineco entfernt nur NTSC spezifische Fehler. Will man die Bilder darüber hinaus noch weiter verbessern, kann man zusätzlich das Programm yuvdenoise einsetzen:

```
> lav2yuv Datei.avi | yuvkineco | yuvdenoise | mpeg2enc -o video.m1v
```

Ein weiteres Programm zur Qualitätsverbesserung von NTSC-Quellmaterial ist das Programm yuvycsnoise.

```
> lav2yuv Datei.avi | yuvycsnoise | mpeg2enc -o video.m1v
```

Das Programm yuvycsnoise entfernt NTSC-typischen separation noise, der auftritt, wenn die Hardware schlechte Y/C-Trennung (luma/chroma) hat. Dies ist erkennbar durch ein Rauschen zwischen hell und dunkel bei vertikalen Bändern (besonders bei rot/blau). Dieses Rauschen wird durch das Programm yuvycsnoise entfernt. Es können natürlich wieder andere Schwellwerte für luma und chroma sowie die Optimierungsmethode eingesetzt werden. Bei digitalem Quellmaterial wird dieses Programm nicht benötigt.

Das Programm yuvycsnoise funktioniert nur richtig, wenn das NTSC-Quellmaterial folgende Eigenschaften hat:

- volle Höhe (480 Linien)
- volle Bildfolgefrequenz ( $29,97 \frac{\text{Bilder}}{\text{s}}$ )
- aufgenommen mit einer Hardware, die eine schlechte Y/C-Trennung hat.

Mehr Informationen über die Programme yuvkineco und yuvycsnoise finden sich im README in dem yufilters-Verzeichnis. Dazu muß man sich das Quellpaket besorgen.

Wenn man die optimalen Einstellungen für die Filter herausfinden will, sollte man in der Befehlskette das Programm mpeg2enc durch das Programm yuvplay ersetzen. Das Programm yuvplay zeigt den Film sofort an, so daß man direkt sehen kann, was die einzelnen Optionen bewirken. Der entsprechende Befehl könnte etwa wie folgt aussehen:

```
> lav2yuv Datei.avi | yuvdenoise -options | yuvscaler -options | yuvplay
```

Es sind uns weiter zwei Firmen bekannt, die Geräte anbieten, die die Qualität des Videos **vor** der Aufnahme verbessern. Dies ist zum einen die Firma SIMA (<http://www.simacorp.com/scc.htm>), deren Gerät aber nur NTSC-Material verarbeitet, und zum anderen die deutsche Firma ELV (<http://www.elv.de>). Bei letzterer gibt es auf dem Internetauftritt der Firma einen shop-Bereich mit einer Sektion Video/Audio-Geräte, wo sich dieses Gerät finden läßt. Die meisten Geräte verarbeiten nur PAL-Quellmaterial.

Als Anhaltswert, wie jedes der einzelnen Filterprogramme die Bitrate beeinflusst, kann die folgende Tabelle dienen. Wird ein Video mit einem Qualitätsfaktor 5 (-q 5) mit variabler Bitrate mit einer maximalen Bitrate von  $8\,500 \frac{\text{kBit}}{\text{s}}$  erzeugt, ergeben sich folgende Beeinflussungen der Bitrate durch die einzelnen Filterprogramme:

- keine Filterung:  $8\,300 \frac{\text{kBit}}{\text{s}}$  (hier wird schon die obere Grenze erreicht)
- yuvdenoise:  $7\,700 \frac{\text{kBit}}{\text{s}}$
- mpeg2enc -reduce-hf:  $7\,400 \frac{\text{kBit}}{\text{s}}$
- yuvdenoise und yuvmedianfilter:  $6\,000 \frac{\text{kBit}}{\text{s}}$
- yuvdenoise und mpeg2enc -reduce-hf:  $4\,900 \frac{\text{kBit}}{\text{s}}$
- Alle Programme zusammen:  $3\,600 \frac{\text{kBit}}{\text{s}}$

Während jedes einzelne Filterprogramm alleine die Bitrate nur wenig verkleinert, reduzieren sie gemeinsam die Bitrate erheblich. Es ergibt nur geringen Unterschied, wenn man anstelle von yuvdenoise dieses Programm zusammen mit mpeg2enc `-reduce-hf` verwendet, wobei gilt, daß der Gewinn an eingesparter Bitrate mit steigendem Qualitätsfaktor immer geringer wird. So verringert man die Bitrate z. B. bei einem Qualitätsfaktor von 4 und der Option `-N 1.0` nur um ca. 1 %, wohingegen bei einem Qualitätsfaktor von 9 und der Option `-N 1.0` bis zu 40 % eingespart werden. Wie groß die Reduzierung der Bitrate durch die Filterprogramme letztendlich ausfällt, hängt sehr stark von der Videoquelle und dem Rauschen des Bildes ab, so daß die hier angegebenen Zahlen nur Anhaltswerte sind. Um bei einem Quellvideo von sehr guter Qualität nach der Kompression jedes Detail wiederzufinden, muß dem Programm mpeg2enc die Option `-keep-hf` oder `-h` mitgegeben werden.

Eine weitere Möglichkeit, die Bitrate zu senken, kann die Option `-E` oder `-unit-coeff-elim` des Programms mpeg2enc sein. Diese in der Voreinstellung ausgeschaltete Option sucht bei ihrer Benutzung mittels eines „unit coefficient elimination“-Algorithmusses das Bild nach Stellen, die nicht viel Information für das Bild liefern, aber zum Speichern viel Bandbreite benötigen, ab. Der typische Wertebereich dieser Option liegt zwischen `-20` und `+20`. Ein negativer Wert bedeutet, daß die Koeffizienten auf null gesetzt werden, während bei positiven Werten die Texture gleich gelassen wird, aber die base intensity coefficients auf null gesetzt werden. Üblich ist, daß durch diese Option die Bandbreite etwa um 5 % gesenkt wird, wobei diese Absenkung aber von fast nicht bemerkbar bis zu 20 % variieren kann.

Wenn man der Meinung ist, daß eine andere Quantisierungs-Matrize das Bild verbessert und die Bandbreite absenkt, kann man bei dem Programm mpeg2enc mit der Option `-K` oder `-custom-quant-matrices` andere als die voreingestellten Matrizen testen. Beim Programm mpeg2enc sind neben der Standardmatrize auch die Matrizen KVCD, tmpgenc, hires sowie eine selbstentwickelte Matrize einsetzbar. Bei Verwendung der Matrizen KVCD und tmpgenc werden die Dateien normalerweise kleiner als bei der Standardmatrize, bei Verwendung der Matrize hires ist dies eher nicht der Fall. Exakte Richtlinien sind schwer aufzustellen, da die Absenkung der Bandbreite sehr stark vom Ausgangsmaterial abhängt. Sie kann reichen von fast unverändert bis zu 20 % kleiner. Größere Absenkungen als 20 % sind eher selten. Man kann bei durchschnittlichen Qualitätsfaktoren (`-q 9 - 11`) mit Absenkungen zwischen 10 - 15 % rechnen, wohingegen bei hohen Qualitätsfaktoren (`-q 4 - 6`) nur einige Bits eingespart werden.

Die Option „unit coefficient elimination“ und die Verwendung anderer Matrizen zur Bildkompression als die Standardmatrize verbessern das Bild, so daß die Bildqualität über der Bildqualität entsprechend dem gewählten Qualitätsfaktor liegt. Man erhält also eine höhere Qualität als die vorgewählte. Interessant ist nun die Frage, welche Dateigrößen sich bei gleicher Bildqualität ergäben, denn dann könnte man gleich eine geringere Qualität vorwählen. Diese Frage ist aber nur sehr schwer zu beantworten.

## 10.2 Skalierung und Offset-Korrektur

Das einfache Skalieren mit dem Programm yuvscaler wurde schon in Kapitel 9 beschrieben, so daß an dieser Stelle darauf verzichtet werden kann.

Beim Skalieren wird ein Teil des Bildes vergrößert oder verkleinert, was durch das Programm yuvscaler erfolgt.

```
> lav2yuv Datei.ali | yuvscaler -I USE400x400+50+100 | yuvplay
```

Hier wird nur ein Teil des Bildes verwendet und auf die Originalgröße hinaufskaliert. Dabei

ändert yuvscaler auch die pixel aspect ratio, wodurch das Bild beim Anschauen mit yuvplay quadratisch wird. Änderte yuvscaler die pixel aspect ratio nicht, sähe das Bild verzerrt aus. Bei einem interlaced Video müssen in der Option -I BildbreitexBildhöhe+Bildbreitenoffset+Bildhöhenoffset die Bildhöhe und der Bildhöhenoffset ein Vielfaches von 4 sein. Bei notinterlaced Videos müssen alle Werte der Option ein Vielfaches von 2 sein. Ein Problem, welches nicht durch einfaches Skalieren gelöst werden kann, ist ein nicht horizontal zentriertes Bild. Zur Zentrierung solcher Bilder dient das Programm y4mshift, denn durch dieses Programm können Bilder nach rechts oder links verschoben werden.

```
> lav2yuv Datei.eli | y4mshift -n 20 | mpeg2enc -f 3 -b 4000 -q 10 video.m2v
```

Dieses Beispiel verschiebt das Bild um 20 Pixel nach rechts. Bei negativen Zahlen hinter der Option -n wird das Bild nach links verschoben. Dabei muß eine gerade Zahl als Argument verwendet werden. Die eingefügten Pixel sind schwarz.

Nun folgt ein Erklärungsversuch, warum manchmal die Bilder nicht richtig zentriert sind und an einer Seite einen schwarzen Rand haben. Der Grund für den schwarzen Rand ist in der Röhre des Fernsehapparates zu suchen. Die Röhre zeigt nämlich nicht das ganze gesendete Bild, weil ein Teil der Röhre mit einem schwarzen Rand überzogen ist, so daß man in diesen Bereichen das gesendete Bild nicht sieht. Aufgrund dieser Ränder erscheint das Bild am Fernsehgerät immer richtig zentriert. Da die TV-Karte des Rechners aber das ganze Bild aufnimmt, sieht man auch das ganze Bild und somit einen eventuellen Offset, was sich in einer Dezentrierung des Bildes bemerkbar macht.

Ein horizontaler Offset ist normalerweise kein Problem der TV-Karte. Er ist eher ein Problem beim Scannen des Films und der Synchronisation mit dem Trägersignal. Die beiden sind dann nicht genau synchron, was man aber am Fernsehgerät auch nicht sieht.

## 10.3 Konversion der Bildfolgefrequenz

Soll die Bildfolgefrequenz z. B. vom PAL-Format auf das NTSC-Format oder vom Film-Format mit  $24 \frac{\text{Bilder}}{\text{s}}$  auf das NTSC-Format mit  $\frac{30000}{1001} \frac{\text{Bilder}}{\text{s}}$  geändert werden, so kann das mit dem Programm yuvfps erfolgen. Dieses Programm kann die Bildfolgefrequenz verringern durch das Weglassen von Bildern sowie die Frequenz erhöhen durch Kopieren und Einfügen von Bildern. Wenn im Header des Films eine falsche Bildfolgefrequenz angegeben wird, so ist es auch möglich, nur den Header im yuv Film zu ändern und den Film ansonsten unverändert zu lassen.

Da die Bilder nur kopiert (replicated) werden, sollte man eine etwaige Filterung zuvor vornehmen. Dann sollte man die Bildfolgefrequenz ändern und danach Skalieren. Bei interlaced Videos sollte man zuerst deinterlacen und erst danach die Bildfolgefrequenz ändern. Beim umgekehrten Vorgehen kann es sonst passieren, daß man beim Erhöhen der Bildfolgefrequenz beim Abspielen des Videos ein merkwürdiges Zucken in den Bildern erhält. Werden PAL-Filme auf das NTSC-Format mit ca.  $29,97 \frac{\text{Bilder}}{\text{s}}$  konvertiert, wird die benötigte Bandbreite um ca.  $\frac{480 \text{ NTSC-Zeilen}}{576 \text{ PAL-Zeilen}}$  kleiner. Wird die Bildfolgefrequenz von PAL auf NTSC (bei  $\frac{24000}{1001}$ ) oder NTSC Film ( $24 \frac{\text{Bilder}}{\text{s}}$ ) geändert, nimmt die Bitrate um etwa  $\frac{480 \text{ Zeilen} \cdot 24 \text{ Bilder s}}{576 \cdot 25 \text{ Bilder s}}$  ab. Wird die Bildfolgefrequenz vor der Filterung geändert, hat das Programm yuvdenoise Probleme, das Rauschen bildübergreifend wegzufiltern, wodurch sich die Bitrate etwas erhöht.

```
> lav2yuv Datei.eli | yuvfps -r 30000:1001 | yuvscaler -0 SVCD | mpeg2enc -f 4 -o video.m2v
```

Dieser Beispielbefehl konvertiert das Quellvideo auf NTSC-Format mit einer Bildfolgefrequenz von ca 29,97  $\frac{\text{Bilder}}{\text{s}}$  bei VCD-Größe.

```
> lav2yuv Datei.eli | yuvdenoise -F | yuvfps -r 24000:1001 | yuvscaler -0 SIZE-720x480 | mpeg2enc -f 3 -b 4000 -q 7 -o video.m2v
```

Hier wird gezeigt, in welcher Reihenfolge die Programme eingesetzt werden sollten. Also zuerst filtern, dann die Bildfolgefrequenz ändern und zuletzt die Bildgröße ändern.

Es kann passieren, daß die Programme yuvscaler und mpeg2enc die TV Norm nicht erkennen. Sollte das passieren, kann die Norm durch die Option -n n/p/s dem Programm mitgegeben werden.

Wenn man weiß, daß nur der Header die falsche Bildfolgefrequenz angibt, so kann man diese im Header mit folgendem Befehl ändern:

```
> lav2yuv Datei.eli | yuvfps -r 25:1 -c | mpeg2enc -f 3 -b 4000 -q 7 -o video.m2v
```

Durch die Angabe der Option -c ändert das Programm yuvfps die Bildfolgefrequenz nur im Header. Mit der Option -r 25:1 teilt man dem Programm mit, auf welche Frequenz der Header geändert werden soll, hier auf 25:1. Die Bildfolgefrequenz muß immer in Bruchschreibweise angegeben werden.

Wenn man weiß, daß der Header falsch ist und man eine andere Bildfolgefrequenz haben will, so kann man das in einem einzigen Schritt durchführen mit folgendem Befehl:

```
> lav2yuv Datei.eli | yuvfps -s 24:1 -r 25:1 | mpeg2enc -o video.m1v
```



# 11 Dekomprimierung von mpg-Filmen

Hat man einen fertig komprimierten Film, den man nachträglich noch verändern und editieren möchte, ergibt sich die Notwendigkeit, die Kompression rückgängig zu machen. Dies ist auf verschiedenen Wegen möglich.

## 11.1 Dekomprimierung mit MPlayer

Die Dekompression von Filmen mit dem Programm MPlayer ist eine gute Methode, jeden Film, den der MPlayer abspielen kann, wieder in ein Format zu wandeln, das mit den mjpeg-tools bearbeitet werden kann. Diese Methode funktioniert mit der Version MPlayer 1.0pre2 oder neuer. Die Befehlsfolge zu Dekomprimierung lautet:

```
> mkfifo stream.yuv
> mplayer -nosound -noframedrop -vo yuv4mpeg Datei.mpg &
> cat stream.yuv > yuv2lav -o mjpeg_wo.avi
> mplayer -vo null -ao pcm -aofile Datei.wav Datei.mpg
```

Jetzt liegt das Video als dekomprimierte avi-Datei `mjpeg_wo.avi` ohne Ton vor. Die Tonspur ist in der separaten Datei `Datei.wav`, die nun mit dem Programm `lavaddwav` der Bildspur hinzugefügt werden kann. Der so dekomprimierte Film kann nun editiert werden (z. B. mit `glav`). Manchmal passiert es, daß der `mplayer` in dieser Reihenfolge das Video nicht decodiert. Hier hilft es, zuerst den `cat` Befehl zu starten und dannach erst die Decodierung bei `mplayer`. Mit `cat` kann man den `wav` File problemlos `mp2enc` fürs Encoding zuführen. Die `-vo yuv4mpeg` Option funktioniert problemlos mit anderen Videoquellen, die in der `mplayer` Dokumentation beschrieben werden.

Statt `yuv2lav` kann man natürlich auch jedes andere Tool wie z. B. `mpeg2enc` von den `mjpegtools` verwenden. Ein Befehl kann daher auch ohne weiteres so aussehen:

```
> cat stream.yuv | yuvdenoise | yuvscaler -o SVCD | mpeg2enc -f 4 -o video.m2v
```

## 11.2 Dekomprimierung mit mpeg2dec

Mpeg2-Filme können mit dem mpeg2dec dekomprimiert werden, vom libmpeg2-Projekt, hier zu finden: <http://libmpeg2.sourceforge.net/>, und das Ausgabeformat pgmpipe zu nutzen. Mit dem Programm pgmtoy4m der mjpegtools kann dann der Ausgabefilm im pgm-Format in das yuv-Format gewandelt werden.

```
> mpeg2dec -s -o pgmpipe ANYTS.VOB | pgmtoy4m -i t -a 10:11 -r 30000:1001 |  
mpeg2enc -f 8 video.m2v
```

Die Dekompression des Tons kann wie im nächsten Kapitel erläutert erfolgen.

## 11.3 Weitere Formatwandlungsmöglichkeiten von mpeg2-Filmen

Beim Transcoding, also Formatumwandlung von MPEG2 Streams von Digital TV oder DVD Quellmaterial, zu MPEG Streams kann man mit einer niedrigeren Bitrate rechnen, als wenn man sie von den üblichen analogen Quellen aufnimmt. Üblicherweise bekommt man bei VCD Größe schon bei 1152 *kBit* vernünftige Qualität bei MPEG1. Den Unterschied macht hier das Signal/Rauschverhältnis des Originals. Das Rauschen bei analogen Signalen macht es wesentlich schwieriger das Video zu kodieren. Eine gute Hilfe für die Umwandlung von Formaten kann unter <http://www.bunkus.org/dvdripping4linux/index.html> gefunden werden. Es werden weiter die Programme ac3dec und mpeg2dec benötigt, die von <http://sourceforge.net/projects/mjpeg> heruntergeladen werden können. Optional kann man für einige Teile auch die Programme sox und toolame verwenden.

Im Script-Verzeichnis findet sich ein Script mit Namen mpegtranscode, mit dem die meiste Konvertierarbeit erledigt werden kann.

Eine Kovertierung kann z. B. so aussehen:

```
> mjpegtranscode -V -o vcd_Film mpeg2src.mpg
```

Hierin bedeuten:

**-V**

Diese Option sagt, daß eine VCD erzeugt wird.

**-o vcd\_Film**

Es werden ein Video mit Namen vcd\_Film.m1v und eine Tondatei mit den Namen vcd\_Film.mp2 erzeugt.

**mpeg2src.mpg**

Name des zu wandelnden Films.

Das Script gibt weiter etwas ähnliches wie

```
> SYNC 234 mSec
```

aus. Weiter muss der Delay von Ton und Bild manuell eingegeben werden, um sicherzustellen, daß beim Multiplexen Ton und Bild synchron sind. Der exakte Wert (in Millisekunden) ist die obige Ausgabe des Scripts. Dieser Wert muß dem Programm mplex mit der Option -O mitgegeben werden.

Der Multiplexbefehl sieht dann folgendermaßen aus:

```
> mplex -f 1 -O 234 vcd_Film.mp2 vcd_Film.m1v -o video.mpg
```

Hierin bedeuten:

**-f 1**

Es wird eine VCD erzeugt.

**-O 234**

Das ist der Offset zwischen Ton und Bild in Millisekunden, wie er vom mjpegtranscoding-Script ausgegeben wurde. Negative Zahlen sind erlaubt.

**vcdFilm.mp2 + vcdFilm.m1v**

Dies sind die vom mjpegtranscoding-Script erzeugten Dateien.

**-o video.mpg**

Das Ausgabevideo als VCD.

Es folgt ein Beispiel zur Erzeugung einer SVCD.

```
> mjpegtranscode -S -o svcd_Film mpeg2src.mpg
```

Die Ausgabedateien des Scripts werden mit dem Befehl

```
> mplex -f 4 -O 234 svcd_Film.mp2 svcd_Film.m2v -o video.mpg
```

zusammengeführt. Manchmal gibt es hier ein Problem beim Wandeln von Filmen im NTSC-Format in eine VCD, weil die Filme u. U. im 3:2 pulldown NTSC-Modus mit  $60 \frac{\text{Bilder}}{\text{s}}$  aufgenommen wurden. Das Programm mpeg2dec wurde aber für die Wiedergabe von Filmen auf Rechnern entwickelt. Es erzeugt Filme mit  $24 \frac{\text{Bilder}}{\text{s}}$ . Wenn man das Video nun komprimiert, wird ein Video mit  $30 \frac{\text{Bilder}}{\text{s}}$  erzeugt, so daß  $\frac{1}{2}$  das Video nun viel zu kurz ist für die komprimierte Tonspur.

Die Umwandlung kann dennoch zum Laufen gebracht werden, aber das muß mit Hand erfolgen mit dem Befehl:

```
> mpeg2dec -s -o pgmpipe mpeg.mpg | pgmtoy4m -i t -a 59:54 -r 25:1 |  
mpeg2enc -I 0 -f 4 -q 9 -V 230 -p -P -o video.m2v
```

In diesem Befehl teilt die Option -p dem Programm mpeg2enc mit, im Header den 3:2 pulldown-Modus für den Film mit  $24 \frac{\text{Bilder}}{\text{s}}$  aufzunehmen. U. U. funktioniert die Wandlung auch ohne die Option -p. Beim Erzeugen einer VCD wird diese Option nicht benötigt, weil sie von mpeg1 nicht unterstützt wird.

Statt mit dem Script mjpegtranscoding kann die Formatwandlung natürlich auch in einzelnen Schritten durchgeführt werden. Die dazu notwendigen Befehlsfolgen werden im folgenden gezeigt.

Zuerst wird der Ton extrahiert.

```
> mpeg2dec -s -o pgmpipe mpeg.mpg | pgmtoy4m -i t -a 59:54 -r 25:1 | yuvscaler  
-O SVCD | mpeg2enc -f 4 -q 9 -o video.m1v
```

Eine der ersten Ausgabezeilen dieses Befehls enthält die SYNC-Angabe des TON/Bild-Offsets, die später beim Multiplexen benötigt wird. Die Angabe `2>/dev/null` im Befehl leitet die Ausgabe von `ac3dec` nach `/dev/null` um. Im nächsten Schritt wird die Tondatei im mpeg-Format erzeugt.

```
> cat Ton.wav | mp2enc -V -v 2 -o Ton.mp2
```

Hierin bedeuten:

**-V**

Erzwingt das VCD-Format. Die Sample-Rate wird von  $44,1\text{ kHz}$  auf  $48\text{ kHz}$  gewandelt.

**-v 2**

Diese Angabe ist nicht unbedingt notwendig. Sie veranlaßt das Programm `mp2enc` auszugeben, wieviele Sekunden der Tondatei schon komprimiert sind.

**-o Ton.mp2**

Benennt die Ausgabedatei.

Der nächste Befehl, der für die Formatwandlung notwendig ist, lautet

```
> cat test2.mpg | extractac3 - -s | ac3dec -o wav | sox -t wav /dev/stdin -r 44100 /dev/stdout | toolame -p 2 -b 224 /dev/stdin Ton.mp2
```

Eine der ersten Ausgabezeilen enthält wieder die delay-Angabe des Ton hinter dem label SYNC. Dieser Wert wird beim Multiplexen gebraucht (SYNC\_value im unten stehenden Multiplexbefehl)

Zum Erzeugen einer VCD dient folgender Befehl:

```
> mpeg2dec -s -o pgmpipe mpeg.mpg | pgmtoy4m -i t -a 59:54 -r 25:1 | yuvscaler -O SVCD | mpeg2enc -f 4 -q 9 -o video.m1v
```

Der Stream von `mpeg2dec` wird wieder durch `pgmtoy4m` in Form gebracht und dann mit `yuvscaler` auf die SVCD Größe gebracht.

Das Multiplexen erfolgt dann mit dem Befehl

```
> mplex -f 1 -O SYNC_value Ton.mp2 video.m1v -o VCD.mpg
```

Es bedeuten:

**-f 1**

Es wird eine VCD erzeugt

**-O SYNC\_value**

Dies ist der oben erwähnte Ton-delay

Andere Ausgabeformate können z. B. mit folgender Befehlskette erzeugt werden.

```
> mpeg2dec -s -o pgmpipe mpeg.mpg | pgmtoy4m -i t -a 59:54 -r 25:1 | yuvscaler -O SIZE_320x200 -O NOT_INTERLACED | mpeg2enc -o anderes_video.m1v
```

Wenn man mpeg-Filme bearbeiten will, so ist das auch möglich, funktioniert aber etwas anders. Für das Demultiplexen kann das Programm bbdmux von dem bbtools-Paket benutzt werden. Dieses Programm trennt entweder das Bild oder den Ton sehr sauber ab. Auf der homepage von Brent Beyler ist es leider nicht mehr verfügbar, es kann aber noch im Internet gefunden werden, wenn die Suchwortkombination bbtools linux suse -blackbox benutzt wird. Gegenwärtig kann es unter <http://www.nop.org/inkling/> gefunden werden.

Zur Anwendung geben Sie zunächst den Befehl

```
> bbdmux video.mpg
```

Als Ausgabe sollte etwas wie folgendes erscheinen:

```
Found stream id 0xE0 = Video Stream 0
Found stream id 0xC0 = MPEG Audio Stream 0
Found stream 0xBE = Padding Stream
Extrahieren Sie nun die Tonspur mit dem Befehl
```

```
> bbdmux video.mpg 0xC0 Ton.mp1
```

Konvertieren Sie die Datei in das wav-Format.

```
> mpg123 -w Ton.wav Ton.mp1
```

Extrahieren Sie das Video mit dem Befehl

```
> bbdmux video.mpg 0xE0 video.m1v
```

Nun wird das Video in das mjpeg-avi-Format konvertiert.

```
> mpeg2dec -o pgmpipe mpeg.mpg | pgmtoy4m -i t -a 59:54 -r 25:1 | yuv2lav -f a
-o video.avi
```

Das Hinzufügen der Tonspur erfolgt nun mit dem Befehl

```
> lavaddwav video.avi audio.wav Ergebnis.avi
```

Wenn das Quellvideo bereits die gewünschte Grösse des Zielvideos hat ist yuvscaler natürlich nicht notwendig. Wenn das Video den Ton im ac3-Format hat, müssen Sie lediglich die u. a. Befehle anwenden.

Tonextraktion:

```
> cat test2.mpg | extractac3 - -s | ac3dec -o 2>/dev/null > Ton.wav
```

Videoextraktion:

```
> mpeg2dec -s -o pgmpipe mpeg.mpg | pgmtoy4m -i t -a 59:54 -r 25:1 | yuv2lav -f
a -q 85 -o video.avi
```

Hinzufügen des Tons:

```
> lavaddwav video.avi Ton.wav Ergebnis.avi
```

Beachten Sie, daß Sie für die Konvertierungen viel Speicherplatz benötigen. Ein Video mit einer Größe von  $1\text{ GB}$  hat im SVCD-Format die Größe von  $2\text{ GB}$ , und natürlich wird noch weiterer Speicherplatz für temporäre Dateien benötigt. Auch benötigt das Konvertieren viel Zeit, haben Sie also Geduld. Weiter muß man beachten, daß bei jeder Konversion Qualität verloren geht.

# 12 Abwägung Qualität gegen Geschwindigkeit

Wenn höchste Qualität das Ziel ist, kann eine kleine Verbesserung mit den Optionen -4 und -2 erreicht werden. Diese Optionen kontrollieren, wie genau das Programm mpeg2enc schlecht aussehende aneinander angrenzende Teile herausnimmt, wenn es mit der Suche beginnt, noch mit 4x4 oder 2x2 pixel-clustern arbeitet und noch nicht mit einzelnen Pixeln. Die Einstellung -4 1 -2 1 maximiert die Qualität, während -4 4 -2 4 die Geschwindigkeit maximiert. Da die statistischen Werte, die das Programm mpeg2enc verwendet, normalerweise ziemlich gut sind, ist der Qualitätssprung nicht besonders groß.

Wenn man den Suchradius (search radius) verkleinert, erhöht sich ebenfalls die Geschwindigkeit. Durch den Algorithmus, der verwendet wird, werden Werte auf den nächsten durch acht teilbaren Wert gerundet. Auf neueren CPUs ist der Geschwindigkeitsgewinn bei Werten unter -r 16 nicht besonders groß aber der Qualitätsgewinn sollte die längere Komprimierungszeit meistens wert sein.

## 12.1 Filme für Software-player und Disk Playback erzeugen

Normalerweise sind Software-player weit flexibler als Hardware-player wie DVD und VCD. Durch diese Flexibilität erreicht man wesentlich bessere Kompressionsraten für die gleiche Qualität. Der Trick dabei ist, Streams zu erzeugen, die einen großen Videopuffer (500 kB und mehr) haben, und variable Bitraten zu verwenden (Option -q des Programms mpeg2enc und -V des Programms mplex). Bei Software-playern funktioniert auch das MPEG Layer 3 (mp3) Audio Format. Ein guter mp3-Encoder wie z. B. das Programm lame erreicht bei einer Bitrate von  $128 \frac{kBit}{s}$  oder  $160 \frac{kBit}{s}$  gleich gute Ergebnisse wie das Format Layer 2 (mp2) bei einer Bitrate von  $224 \frac{kBit}{s}$ .

# 13 SMP und verteiltes Komprimieren

Die Option `-M/--multi-thread [0...32]` des Programms `mpeg2enc` bestimmt, wie gleichzeitig auszuführende Prozesse verteilt werden. Dadurch kann die CPU-Zeit mehrerer Rechner optimal durch das Programm `mpeg2enc` genutzt werden. Schon in der Voreinstellung (default `-M 1`) werden durch das Programm `mpeg2enc` mehrere Threads gleichzeitig ausgeführt, wie z. B. das Lesen der Frames und das Komprimieren. Mit der Option `-M` kann nun die Kompressionsarbeit auf mehrere CPUs verteilt werden, ohne daß spezielle Pufferprogramme benötigt werden. Wenn man auf einem Rechner mit nur einer CPU komprimiert und man nur wenig Hauptspeicher zur Verfügung hat, hilft zur Optimierung der Kompression das Abstellen der gleichzeitigen Abarbeitung mehrerer Prozesse durch die Option `-M 0`.

Bei SMP-Rechnern (symetric multiprocessing) mit zwei oder mehr Prozessoren kann der Programmablauf von `mpeg2enc` beschleunigt werden, wenn man angibt, wieviele Threads gleichzeitig laufen sollen, z. B. mit der Option `-M 2` für zwei Threads. Die Option `-M 2` oder `-M 3` auf einem Rechner mit zwei Prozessoren sollte den Kompressionsprozeß um ca. 80 % beschleunigen. Werte über 3 sind zwar erlaubt, bringen aber nur mehr wenig Verbesserung auch auf 4 CPU Systemen.

Sollte man eine richtig schnelle SMP Maschine (Stand 1.Aug.03) wie etwa eine dual Athlon MP 2600 oder etwas vergleichbares haben, kann es passieren, daß beide CPUs trotz `-M 2` und `filtern` nicht ausgelastet sind. Hier kann das Programm `buffer` oder `bfr` helfen, in Kombination mit 10 – 20 Mb Puffer beide CPUs voll zu beschäftigen.

Allerdings kann eine Befehlskette zum Komprimieren eines Films mit Filterung des Videos auch schon zwei oder mehr Prozessoren beschäftigen, ohne daß man die Option `-M` benutzt. Entauschen mit `yuvdenoise` oder `yuvmedianfilter` benötigt viel Rechenleistung, fast soviel wie das Komprimieren selbst. Speziell das Programm `yuvmedianfilter` benötigt sehr viel Rechenleistung.

Wenn man mehrere Rechner hat, kann man das Komprimieren und Filtern auf diese Rechner aufteilen mit dem Standard „`rsh`“ oder „`rcmd`“ `r`mote shell execution commands. Für die Verteilung der Arbeit auf zwei Rechner könnte der Befehl dazu etwa wie folgt aussehen.

```
> rsh Rechner1 lav2yuv "Datei.eli | yuvscaler -O SVCD | yuvdenoise"| mpeg2enc -f 4 -o video.m2v
```

Der Rechner, auf dem dieser Befehl eingegeben wird, führt nun nur die eigentliche Komprimierung (`mpeg2enc`) aus. Der Rechner mit dem Namen `Rechner1` macht den Rest der Arbeit wie `Decoding`, `Scaling` und `Denoising`.

Natürlich muß der Rechner1 Zugang zum Video haben. Außerdem muß genügend freier Speicherplatz auf dem Rechner, auf dem der Befehl gestartet wird, vorhanden sein. Es empfiehlt sich, den Speicherplatz für alle Nutzer des Rechners zugänglich zu machen, wie es z. B. `NFS` und andere Pakete machen. Wenn man drei Rechner zur Verfügung hat, kann man die Arbeit



auf drei Rechner verteilen. Dabei kann einer z. B. das Dekodieren und Skalieren übernehmen, der zweite das Entrauschen und der dritte das eigentliche Komprimieren. Der Befehl dazu könnte wie folgt aussehen.

```
> rsh Rechner1 "lav2yuv Datei.eli | yuvscaler -O SVCD" | yuvdenoise | rsh Rechner3  
mpeg2enc -f 4 -o video.m2v
```

Man muß dabei auf die richtige Reihenfolge der Kommandos achten, damit die Daten direkt von der Maschine, die sie sendet, zu der gehen, die sie empfangen soll. Außerdem muß das Netzwerk schnell genug sein, damit es nicht zum Engpaß wird. Es sollte schon ein 100 *MBit* Ethernet Netzwerk sein.

Bei richtig schnellen Maschinen muß es schon ein 100 *MBit* Switched Ethernet (oder besser) sein. Wie man den rshd (Remote Shell Daemon), den man für rsh benötigt, und rsh selbst einrichtet, damit es funktioniert, ist nicht Gegenstand dieses howtos. Da es aber ein übliches Paket ist, sollte es ohne Probleme installiert und aktiviert werden können bei allen gängigen Linux Distributionen.

Man sollte dabei unbedingt darauf achten, daß man bei der Benutzung von rsh keine Sicherheitslöcher schafft. Dies ist vor allem wichtig bei Rechnern, die an öffentlichen Netzen hängen.

## 14 Kontakt und Danksagung

Bei Problemen und/oder Vorschlägen kann man mir (Bernhard Praschinger) gerne eine mail schreiben. Meine Mailadresse lautet <waldviertler@users.sourceforge.net>.

Viele Hinweise habe ich aus dem HINTS-File eingebaut, das von Andrew Stevens (<andrew.stevens@mnet-online.de<) stammt. Danke auch noch an die Leute, die das Manuskript immer wieder überprüft und Fehler ausgebessert haben. Hier haben mir immer wieder Wolfgang Gller bei der deutschen und englischen Version, Steven M. Schultz bei der englischen Version geholfen.

Horst Schröder konnte mich überzeugen, daß sich die Deutsche Version in  $\LaTeX$  auch ganz gut manchen würde. Er hat die vorherige Version nach  $\LaTeX$  transferiert und dabei das Konzept noch umgedreht, das jetzt einsteigerfreundlicher sein sollte.

Derek Fountain hat das Flussdiagramm gezeichnet.

Und dann gibt es da noch einige Leute, die mich mit Informationen zu den Programmen und allgemeinen Hinweisen versorgt haben, auch ihnen **Danke**.