



Tux goes Cinema

High quality image processing for
cinema-like videos (and perfect fotos)

Dipl.-Ing. Dr. Robert Matzinger

Technik/Philosophie:

```
sed '1,$ s/hightech/lowtech/'
```

Übersetzung:



low tech
statt
high tech

Die Geschichte dieses Vortrags

- Es war einmal...
... ein Fotograf, der Videos drehen
wollte ...



Die Geschichte dieses Vortrags

- Es war einmal...
... ein Fotograf, der Videos drehen
wollte ...
- Ziel:
 - Feature Videos, Videoprojekte
 - Tolle Bilder!
- Nichtziele:
 - Schnelle Publikation
 - Profit

Cinematografic Video

- ... “Cinema Quality” accurately
describes a camera’s shallow depth of
field, large sensors, and ability to use
interchangeable lenses ...

<http://crewscontrol.com/index.php/blog-central/shooting-with-digital-latitude-s-log-log-c-luts>

- Charakteristikum:
 - Nachbearbeitung der “Bilder” ist fixer
Bestandteil des Arbeitsprocesses
 - “Grading”
 - Roh-Videos; oft:
Exposure to the right, Log-Curves

Color Grading



Common DSLR Shot

RAW Wide Dynamic Range

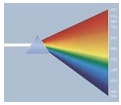
Final Color Graded Shot

[Quelle: Blackmagic]

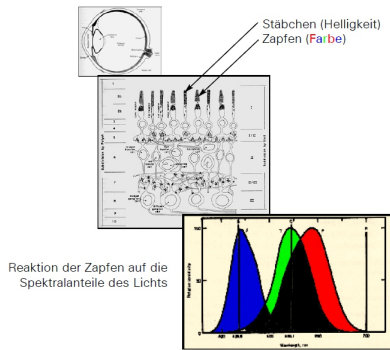
DEMO: Brücke

Grundlagen

• Das menschliche Auge und die Farben

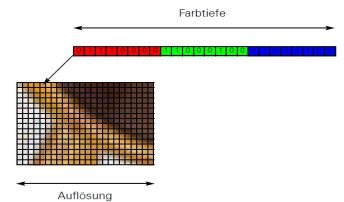


- Spektralfarben (Huygens, 17. Jhdt.)
- Auge:
 - Begrenzte Anzahl von Rezeptoren
 - Farbempfindlichkeit in 3 Bereichen



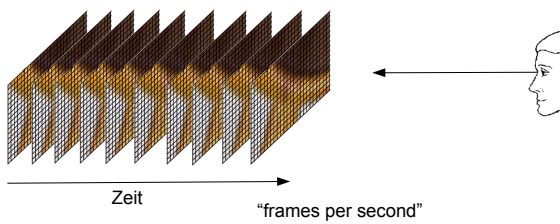
Grundlagen

- Pixelung
- RGB – Red Green Blue
 - Kodiere Helligkeit der drei Farbkomponenten



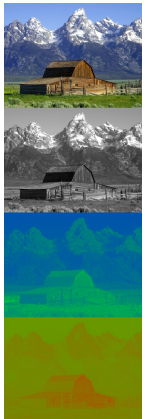
Grundlagen Film

• Bewegte Bilder



Farbcodierung - Pixelformat

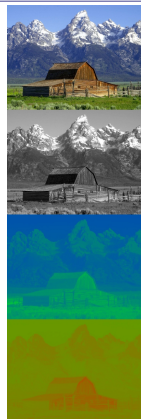
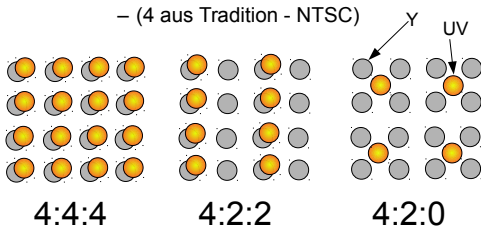
- YUV (statt RGB)
 - Y ... Luminance ... Helligkeit
 - UV ... Chrominance ... Farbe
 - Definierte Umrechnung
 - Kommt vom Schwarzweissfernsehen -
 - Farbe kam später dazu



<http://de.wikipedia.org/wiki/YUV-Farbmodell>

Farbcodierung - Pixelformat

- Das Auge kann Farbinformation nicht so scharf "empfangen" wie Helligkeit
- speichere Farbinformation mit niedrigerer Auflösung
- Abtastrate – Verhältnis
 - (4 aus Tradition - NTSC)



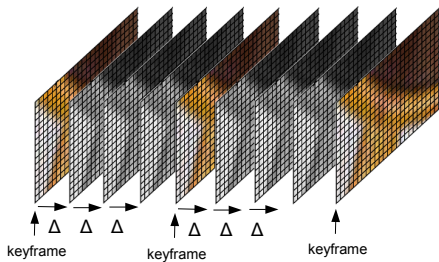
Bildqualität

- Compression Level des "Frame"
 - z.B. JPG
 - Wäge Bildqualität gegen Speicherplatz



Keyframes, I-Frames

- Keyframe: Ganzes Bild
- I-Frame: Übertrage nur Unterschiede



Interlacing/progressive

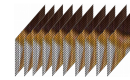
- 1920i ... interlaced video



Gerade/ungerade
Zeilen
abwechselnd



- 1920p ... progressive



Ein Bild nach
dem anderen



Nomenklatur

- Container: mov, mp4, avi, flv, ...
- Codec: Codierung des Video
 - Mpeg-1
 - Mpeg-4
 - Dirac
 - DivX
 - H.264
 - Apple Prores
 - D9
 - AVC-Intra
 - ...



Beispiel/Typisch:
H.264 in MP4 Container

http://de.wikipedia.org/wiki/Liste_der_Videocodecs

Codes für Roh-Videos

... und WIP

Hohe Auflösung, keine B-Frames, hohes
Compression Level, progressive, ...

- RawDNG
 - 16bit Raw Files in Folder
- 10bit Prores, prores yuv422p10le
- Mjpeg, mjpeg yuvj420p ← Cineerra
 - JPG-Files in Container
- ...

Codes für Roh-Videos

... und WIP

- RawDNG
 - 16bit Raw Files in Folder
 - ~2.7 gigabytes per minute == 23 min auf 64GB
- 10bit Prores, prores yuv422p10le
 - ~1.2 gigabytes per min == 53 min auf 64 GB
- Mjpeg, mjpeg yuvj420p
 - JPG-Files in Container
 - ~0.3 gigabytes per minute
- ...

Hardware - Cameras



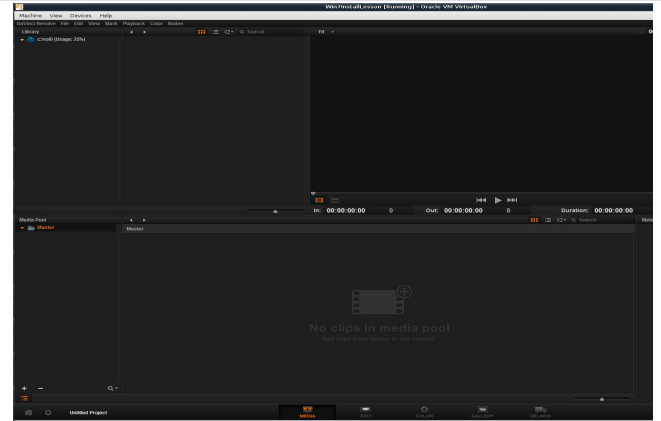
Cinematographic Grading sinnvoll

Die "Profi"-Tools

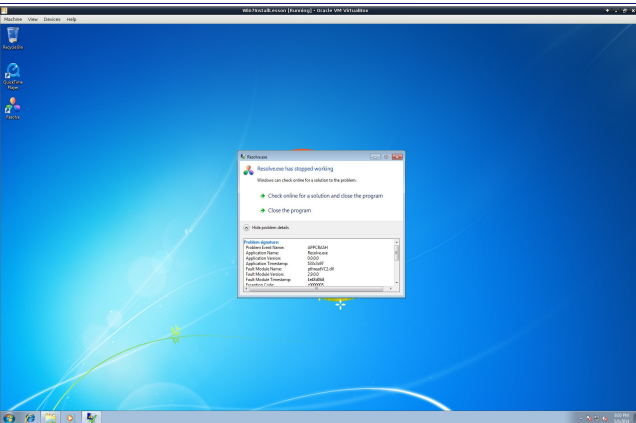
- Filmlight ???
- Adobe Speedgrade
 - ca. € 1200,-
- The Grading Sweet + Adobe Final Cut Pro
 - ca. € 700,-
- Blackmagic DaVinci Resolve
 - Lite Gratis Version – nur Win&Mac, spezielle Hardware?
 - ca. \$ 1000,- PLUS HARDWARE (auch für RHEL Linux; nur!)
 - Station: \$ 30000,-



Die "Profi"-Tools – DaVinci Resolve



Die "Profi"-Tools – DaVinci Resolve



DEMO: pdf

Disclaimer

- Die vorgestellte Methode ist insgesamt NICHT als Konkurrenz zu den "Profi"-Tools gedacht
- → andere Dimension
- Hier gesucht: Kostengünstige Lösung für (ambitionierte) Amateure
- Weniger Features (aber die notwendigen)

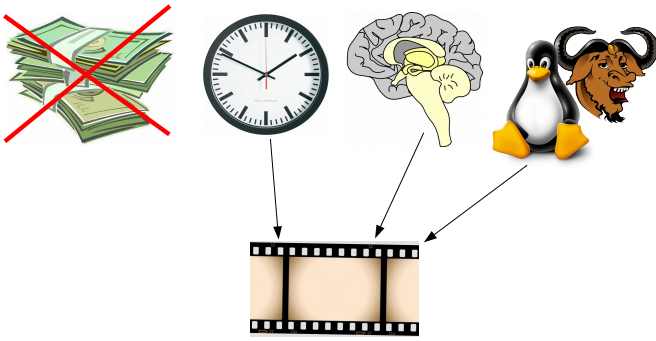
Ressourcen



Ressourcen




Ressourcen



Idee: Command-Line und Scripts

- HERVORRAGENDE TOOLS
- Reproduzierbar
- Archivierbar
- Longzeitnutzen
- Mein Wissen bleibt (mir) zugänglich
- Rendering-„Farm“ leicht möglich
 - ssh-Verbindung reicht aus
 - Überwachung sehr leicht (GNU Screen)

Software

-  **FFmpeg**
 - Das „Swissknife“ der Video-Bearbeitung
 - UNGLAUBLICH viele Filter und Möglichkeiten
 - Auf Videos deutlich schneller als die Bild-Tools
 - schlecht dokumentiert – ausprobieren kostet sehr viel Zeit
 - keine echte 48bit-Umrechnung
 - Meldung
“16bit Chroma Interpolation not yet implemented”
 - Hald-CLUTs funktionieren nicht

Software

- ImageMagick
 - DER Klassiker der Command-Line-Bildbearbeitung
 - Unglaublich viele Möglichkeiten und Optionen
 - Hald-CLUTs
 - 48-bit Arithmetik
 - Unübersichtlich, schwer zu lernen
 - Mäßig dokumentiert
 - Langsam



Software

- GraphicsMagick
 - Fork von Imagemagick
 - weniger Möglichkeiten
 - SCHNELL, guter Batch-Modus
 - Hald-CLUTs
 - 48-bit Arithmetik nur wenn man es selbst aus den Source-Code compiliert
 - Etwas übersichtlicher
 - Mäßig dokumentiert



Software

- GMIC
 - Völlig anderes Konzept als Imagemagick
 - Floating Point Operations (≥ 48 bit)
 - DIE BESTEN NOISE-Reduction und Schärfungs-Algorithmen
 - Wissenschaftlich fundiert...
 - Viele Tricks möglich
 - Gutes Handbuch, übersichtlich
 - Keine Hald-CLUTs



Software

- OpenColorIO
 - Konversion von LUTs
 - DCRAW
 - Extraktion von RAW-Files
 - Bash-Scripting
-
- Geeqie
 - GIMP
 - ffmpeg, vlc, ...

Installation

```
apt-get install opencolorio-tools # Installed: 1.0.8-dfsg0-2.1+b1
apt-get install imagemagick      # Installed: 8:6.7.7.10+dfsg-1
apt-get install gmic              # Installed: 1.5.7.1-4+b1

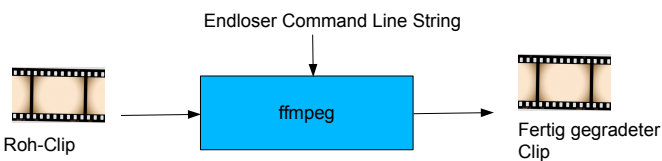
nano /etc/apt/sources.list
http://www.deb-multimedia.org/ jessie/main amd64 Packages
apt-get install ffmpeg

apt-get install build-essential
apt-get source graphicsmagick
cd graphicsmagick-1.3.18
nano debian/rules
## find line: ./configure $(gm_confflags) \
## add:
# --with-quantum-depth=16 \

apt-get build-dep graphicsmagick
dpkg-buildpackage
dpkg -i ../graphicsmagick_1.3.18-2_amd64.deb \
../libgraphicsmagick++3_1.3.18-2_amd64.deb \
../libgraphicsmagick3_1.3.18-2_amd64.deb
```

Workflow?

- Ideal:



- Leider:

- Keine 48bit-Verarbeitung
- Keine Hald-CLUTs

Warum 48 bit Farbtiefe?

- Steigung der Gradationskurve vermindert relative Auflösung

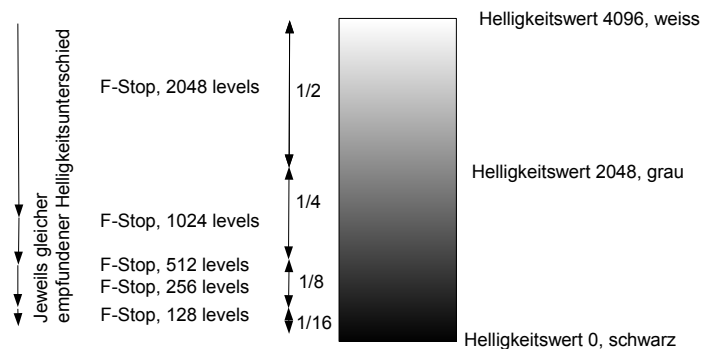
- DEMO image, histogram
- DEMO video, skye

ETTR und LOG-Curves

- Grundproblem
Helligkeit Bildschirm/Sensor/Projektor
Linear
- Helligkeitsempfinden des Auges
Logarithmisch
- Halber Helligkeitswert == ein F-Stop
(Blende) dunkler

ETTR und LOG-Curves

- Beispiel: 12 bit ... 4096 Helligkeitsstufen

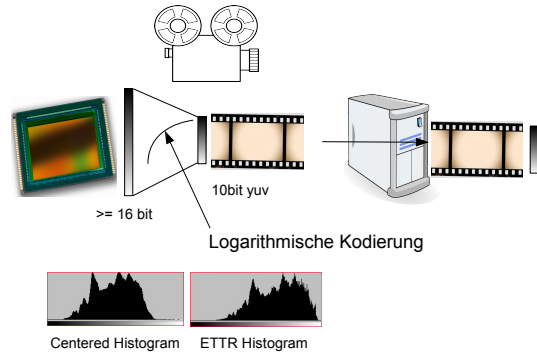


ETTR und LOG-Curves

- Problem: Gradation in dunkeln Bildbereichen
- Problem: Signal-to-noise-Ration in dunklen Bildbereichen
- Problem: Aufhellen von dunklen Bildbereichen → Banding
- Abhilfe: "Exposure to the right" == ETTR LOG, LOG-C
- CINEMATOGRAPHIC

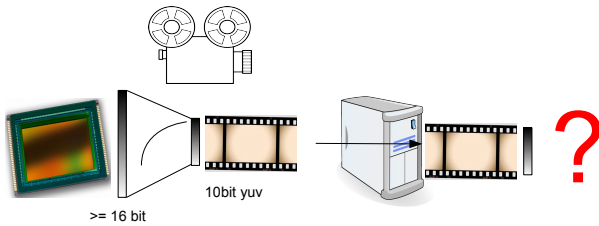
ETTR und LOG-Curves

Logarithmische Kodierung, Exposure to the right

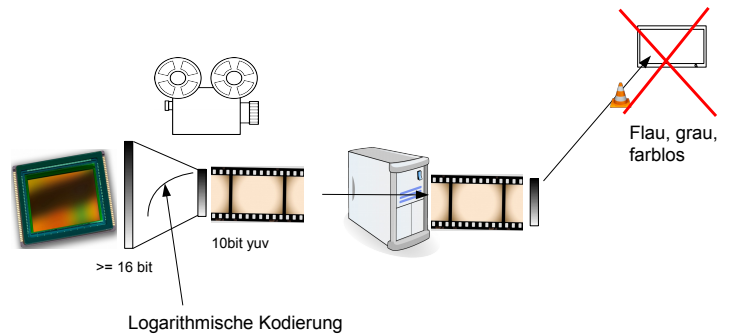


ETTR und LOG-Curves

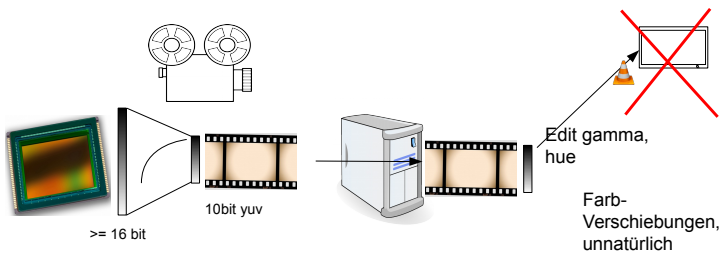
Logarithmische Kodierung, Exposure to the right



ETTR und LOG-Curves

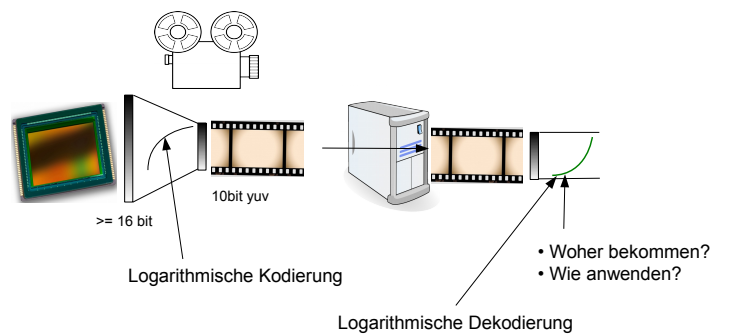


ETTR und LOG-Curves

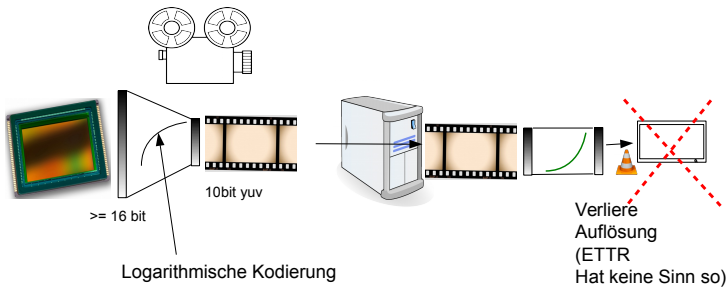


DEMO

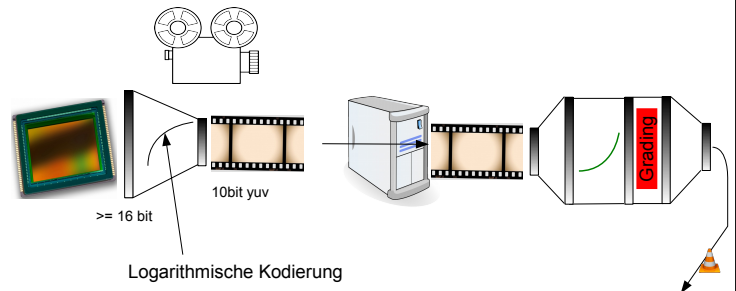
ETTR und LOG-Curves



ETTR und LOG-Curves



ETTR und LOG-Curves



Log-Curves

- LOG-Kodierung “Log”, “LogC”, “ArriLog”, “Cinema-style”, ...
herstellerabhängig
- Dekodierung, Color-Lookup-Table
Woher?
 - Arri ... Download
 - Sony ... Download
 - Blackmagic ... extract from software
 - Amateur Sites – DonationWare

LINKS

LUT

- Color LookUp Table
-
- Speichere Abbildung $(RGB)_{alt} \rightarrow (RGB)_{neu}$ als 3-dimensionale Tabelle dazwischen Interpolation
 - Beliebige Color Conversions codierbar

LUT Formate

- .dat, .cube, .3dl, ...
- Conversion

```
ociobakelut --format flame --lut CameraFilmToRec709.cube CameraFilmToRec709.3dl
```

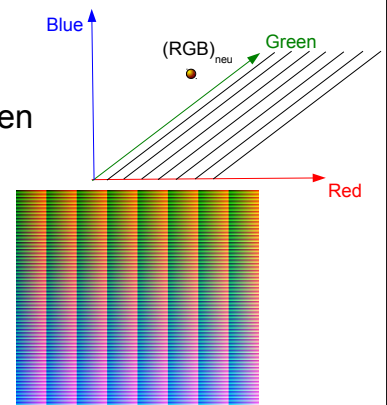
- Einsatz in ffmpeg

```
ffmpeg -i ... -vf "lut3d=CameraFilmToRec709.3dl,..." ...
```

- ABER: keine 48-bit Konversion
 - Not yet implemented
 - ffmpeg richtet sich immer nach dem letzten Format in der Konversions-Kette

Hald-CLUT

- Schneide Tabelle “in Scheiben”
- Nebeneinanderlegen \rightarrow Bild
- Hald-CLUT
- Codiert eine Farb-Konversion



Hald-CLUT

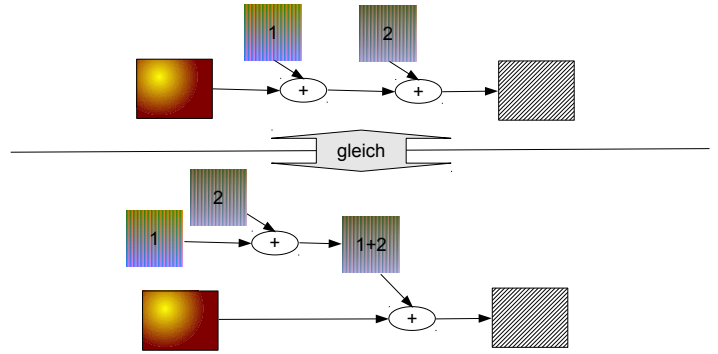
- Imagemagick und GraphicsMagick verstehen Hald-CLUTs
- Conversion aus .3dl-Files mit ffmpeg

```
ffmpeg -f lavfi -i haldclutsrc=8 \
-vf lut3d=CameraFilmToRec709.3dl:interp=trilinear \
-frames:v 1 -f image2 -pix_fmt rgb24 CameraFilmToRec709CLUT.png
```

Meine Hald-CLUT!

Hald-CLUTs

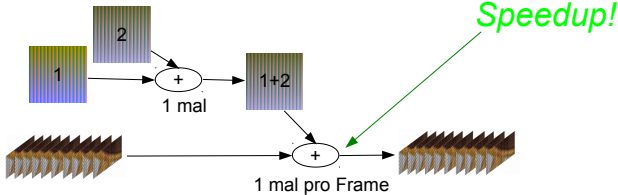
- Hald-CLUTs lassen sich stapeln



Hald-CLUTs

- Kombination von Hald-CLUTs

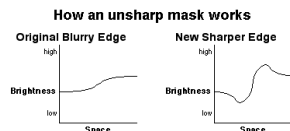
- 1 Camera-LUT
- 2 Eigenes (beliebiges) Grading



Was noch?

Sharpening

- Verstärkte Farbübergänge – unsharp masking



- Gibt es in ffmpeg, imagemagick, gmic, ...
- Verbesserungen möglich:
 - Deconvolution
 - "GoldMeinel"-Deblur (gmic)

Noise Reduction

- Ursache: Sensor-Rauschen
- Je wärmer der Sensor, desto mehr rauschen
- Sensorkühlung!!!
- Noise Reduction wird oft schon in der Kamera durchgeführt.
- ABSCHALTEN!!!!

Noise Reduction

- Verschiedene Strategien möglich.
- Einfache Noise Reduction
 - Blur – Bild unschärfer machen.
 - Adaptive Blur – nur in kontrastarmen Bereichen Bild unschärfer machen.
 - (Adaptive) Chroma Blur – Nur die Farbinformation wird unschärfer gemacht

- Problem: Puderquasten-Effekt

DEMO!

Noise Reduction

- Komplexe Methoden
 - Wavelet denoiser
 - `ffmpeg ... -vf owdenoise=depth=8:ls=5.0:cs=5.0`
 - `gmic ... -denoise_haar ...`
 - 2D DCT frequency domain filtering
 - `ffmpeg ... -vf dctdnoiz=s=2.5:overlap=8`
 - Shape Adaptive Blur
 - `ffmpeg ... -vf sab=lr=0.9:lpfr=1:ls=3:cr=0.9:cpfr=1:cs=3`
 - Anisotropic smoothing using diffusion PDE's
 - `gmic -smooth 30,.86,0.34,0.6,1.1,0.8,30,2,1,1 ...`

Noise Reduction

- - Wavelet denoiser Super, aber einfach zuuuuu langsam
 - `ffmpeg ... -vf owdenoise=depth=8:ls=5.0:cs=5.0`
 - `gmic ... -denoise_haar ...`
 - 2D DCT frequency domain filtering Für heikle Videos, langsam
 - `ffmpeg ... -vf dctdnoiz=s=2.5:overlap=8`
 - Shape Adaptive Blur Für alle Videos, Gute Kompromiss Tempo/Qualität
 - `ffmpeg ... -vf sab=lr=0.9:lpfr=1:ls=3:cr=0.9:cpfr=1:cs=3`
 - Anisotropic smoothing using diffusion PDE's DEMO paper
 - `gmic -smooth 30,.86,0.34,0.6,1.1,0.8,30,2,1,1 ...`

Wissenschaftlich fundierte Methode. Für Fotografie. **SENSATIONELL!**

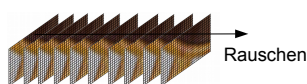
DeShaking

- Eliminiere Kamera-Zittern
 - Ähnlichkeitsrechnung zwischen Frames
 - 2 Passes
 - `ffmpeg -vf "...vidstabdetect..."`
 - `ffmpeg -vf "...vidstabtransform..."`
- Besser: Stativ
 - Unschärfe im Frame durch Kamerazittern kann nicht behoben werden
 - 25 fps, 180° Shutter Angle == 1/50 sec Belichtungszeit

Demo

Timeline Noise Removal

- Rauschen einzelner Pixel in der Zeitachse



- `ffmpeg -vf "... hqdn3d=luma_spatial=5 ..."`

Mein Workflow in der Praxis

- Zentral: VIDEO-Utility (bash)
- Für jedes Video ein Steuer-Script
 - Beliebige Chain konfigurierbar!,
 - eventuell `default.vs`
 - Default Grading über automatische Auswertung der Histogramme

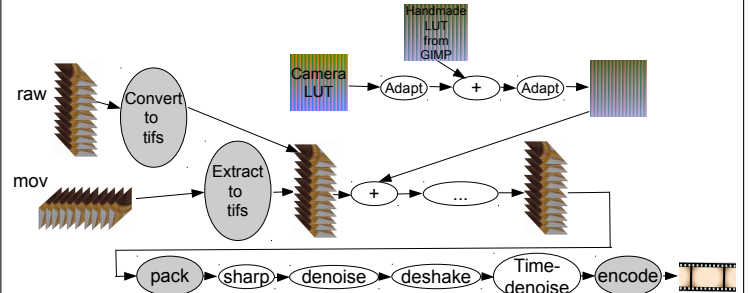
Demo

Mein Workflow in der Praxis

- 1. Beispielframes aus Video extrahieren
 - 1a. Histogramm-Grenzen berechnen
- 2. Grading als Hald-CLUT speichern
 - Editiere Beispielframes mit GIMP
 - Gleiche Bearbeitung auf identity-Hald-CLUT
 - *Alternativ (oder ergänzend) AUTO-Histogramm*
 - Anwenden
- 3. Weitere Schritte adaptieren
- 4. Auf Beispielframes anwenden
 - ev. zurück zu 2.
- 5. Grading-Job ... läuft auf Server im Hintergrund

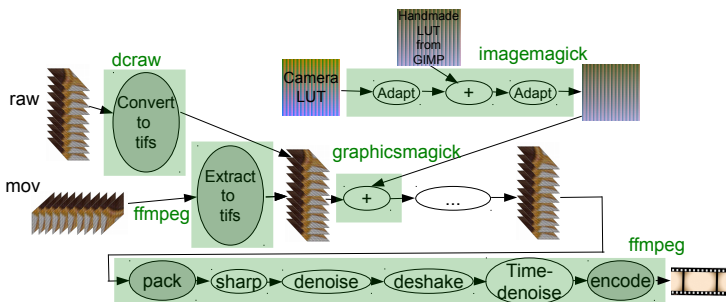
Video-Grading Chain

- Zentral: VIDEO-Utility (bash)
- Für jedes Video ein Steuer-Script (beliebig)
- Übliche Chain:



Video-Grading Chain

- Zentral: VIDEO-Utility (bash)

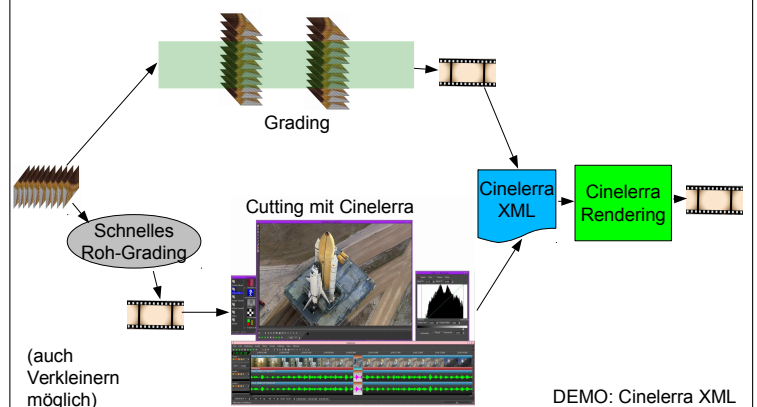


DEMO

Grading/Cutting Parallelisierung

- Mit (z.B.) Cinelerra:
- Parallelisierung von
 - Grading ... Video-Script, GIMP, ...
 - Cutting ... Cinelerra möglich!

Grading/Cutting Parallelisierung



Fazit

- Bedienung --
sperrig, aber machbar
- Vorteile (von Scripts): Archivierbar,
Kopierbar, Versionierbar
- Remote Grading, Cluster leicht: ssh
- Qualität +++
Die besten Algorithmen, die es gibt!
- Ausbaufähig, adaptierbar

Geplant:

- Verläufe
- Tricks:
 - Anfangssequenz
 - Reverse, Zeitraffer, ...
 - Titel, etc.als fertige Scripts
- Masken, etc.

Fazit: Die Features sind da, es fehlen nur
Scripts, die die richtigen Parameter
beisteuern.

Zum Schluß

- Demo
- Kontakt:
<http://www.fh-burgenland.at/kontakt/team/team/m/>
- Scripts: Auf Anfrage

Mind the Command Line!