3D-Videoaufzeichnung mit der Webcam?

Günter Pomaska, gp@imagefact.de

Eine Webcam oder im Zusammenhang mit der hier vorgestellten Anwendung besser als PC-Kamera oder auch USB-Kamera bezeichnet, liefert über die serielle Schnittstelle kontinuierlich zeilenweise Bilder an den PC. Durch entsprechenden Treiberprogramme werden die Datenstreams in Bilder umgesetzt und gespeichert. Videoaufzeichnung mit 1080p im Format 16:9 bei 30 fps gelten mittlerweile als Standard für Kameras der oberen Güteklasse. Plattformübergreifende Software zur Videoaufzeichnung sind neben anderen das Multimedia-Framework FFmpeg und der VLC Player. Aber auch VirtualDub ist unter Windows immer noch ein beliebtes Werkzeug für Aufnahme und Bearbeitung von Videodaten. Programmierer werden sich mit OpenCV anfreunden. Der vorliegende Beitrag behandelt u.a. die Möglichkeiten des Parallelbetriebs zweier Kameras mit einem Rechner und Hinweise zur Synchronisation und Ausrichtung der Videoclips.

Anforderungen an dynamische 3D-Aufnahmen

Innerhalb einer Bildperiode werden die Pixel eines Bildes mit einer Webcam zeilenweise belichtet. Die Bildfrequenz, also Dauer der Bildaufnahme, wird in Frames per Second (fps) angegeben und ist kameraabhängig auch mit der Auflösung korreliert. Wir haben es wie üblicherweise bei CMOS-Sensoren mit einem Rolling Shutter zu tun, demgegenüber ein Global Shutter den gesamten Frame zeitgleich aufnimmt. Elektronische Einrichtungen zur Synchronisierung, wie etwa der bekannte Lanc-Controller für Sony Videokameras, sind für Webcams, abgesehen von individuellen Lösungen, nicht verfügbar. Wir können also nur versuchen, die Kamerastreams zeitgleich zu starten oder im Schnittprogramm zu synchronisieren. Des Weiteren sind noch die geometrischen Bedingungen der Stereoskopie, insbesondere die Vermeidung von Vertikalparallaxen zu berücksichtigen. Gefordert sind identische Einstellungen (Controls) u.a. für Belichtung, Fokus und Brennweite der beiden Halbbilder. Auch wenn verschiedene Einflüsse bei der Videobetrachtung zunächst unauffällig erscheinen, sollte man um deren Beseitigung bemüht sein. Zumal bei längerer Betrachtung Störungen zu Unwohlsein führen können. Aber einen abendfüllenden Film wollen wir mit der Webcam ohnehin nicht drehen.

Warum sollte man nun PC-Kameras für die Stereoskopie nutzen? Gerade im Nahbereich bieten sich da Vorteile aufgrund der Fokussierung ab einem Bereich von nur wenigen Zentimetern an, Aufgrund der kompakten Bauweise ist eine kurze Basis zu realisieren. Weiter sind auch die Anforderungen an die Lichtbedingungen gering. Auch wäre die Netzwerkeinbindung eine Option. Für Versuchsaufbauten kann man preislich im einstelligen Eurobereich einsteigen. Nachfolgend kommen

zwei PC-Kameras Microsoft LifeCam Studio mit einer Auflösung von 1080p und einer Bildwiederholrate von 30fps zur Anwendung. Das Bildformat ist 16:9. Die LifeCam von Microsoft gehört zum oberen Qualitätssegment der PC-Kameras. Die Ausrichtung der Kameras wird durch einen im 3D-Drucker hergestellten Kamerarig, der Basisabstände von 33 mm bis 90 mm unterstützt, näherungsweise parallel vorgenommen. Eine präzise Justierung bleibt der Nachbearbeitung vorbehalten.

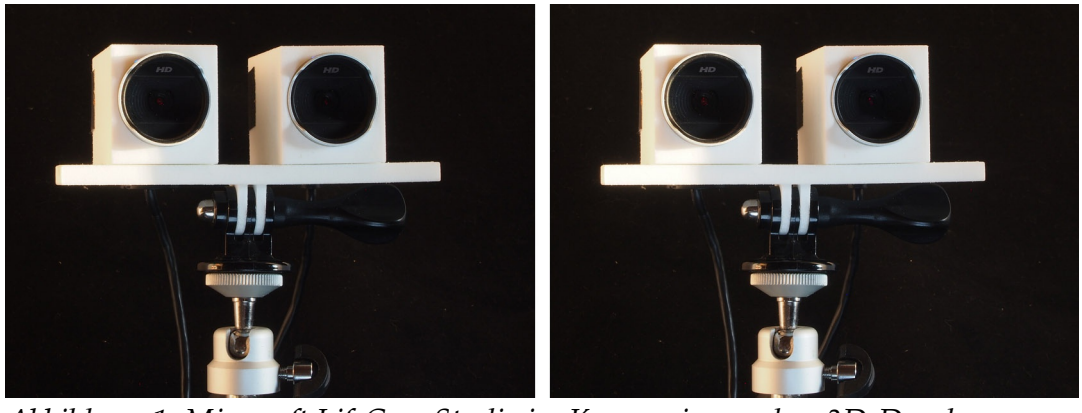

Abbildung 1: Microsoft LifeCam Studio im Kamerarig aus dem 3D-Drucker

Webcams unter Windows mit dem VLC Media Player

Der VLC Media Player unter Windows (läuft auch unter Linux) dient als Aufzeichnungssoftware. Bei ca. 33 mm Aufnahmebasis kann eine Nahentfernung von etwa 60 cm einen hinreichenden 3D-Eindruck liefern. Einstellungen für Zoom, Fokussierung u.a. sind im Kameramenü der Software vorzunehmen. Somit ist die Gleichheit der Aufnahmeeinstellungen für beide Kameras gewährleistet. Zunächst gehen wir davon aus, dass mit einem Rechner die Bandbreite für Bildauflösung und Wiederholungsrate hinreichend ist. Die Konfiguration mit einer zweiten CPU schließen wir bei einem derartig minimalistischen Konzept aus.

Vergleichen Sie zu den folgenden Ausführungen auch die Screenshots in den Abbildungen 2 und 3.Wir starten den VLC Media Player für die linke Kamera. Mit *Medien > Aufnahmegerät öffen* wird im Aufnahmemodus *DirectShow* das Videogerät mit *Erweiterte Optionen (*Checkbox Geräte-Eigenschaften anklicken) ausgewählt und die Wiedergabe gestartet. Im sich öffnenden Panel *Eigenschaften* belassen wir es zunächst beim Standard und gehen in die Registerkarte *Kamerasteuerung*. Auch hier akzeptieren wir die Standardwerte, können aber später die Einstellungen für gleiche Bedingungen beider Kameras nutzen. Nach Bestätigung der Eingabe meldet sich ein weiteres Panel mit dem *Datenformat*.

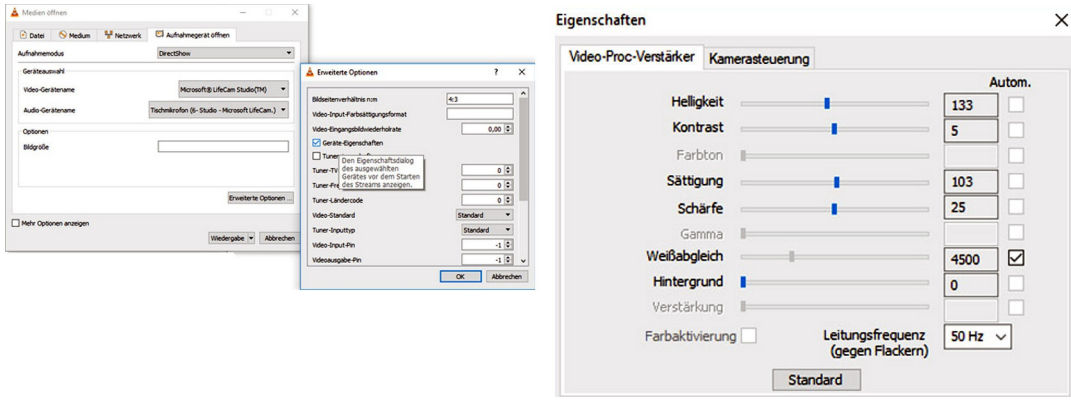

Abbildung 2: Auswahl des angeschlossenen Geräts und Kamerasteuerung

Im Menü *Videoformat* kann nunmehr die Größe der Abbildung und das Seitenverhältnis angepasst werden. Übernehmen Sie ggf. die angebotenen Daten oder setzen Sie die Auflösung und Framerate etwas runter um der Bandbreite des Systems gerecht zu werden.Der Zielpfad für die zu speichernde Videodatei wird unter *Werkzeuge > Einstellungen Registerkarte Eingang/Codecs* angegeben. Jetzt wird noch der rote Recordknopf benötigt. Den aktivieren Sie unter *Ansicht > Erweiterte Steuerung*.

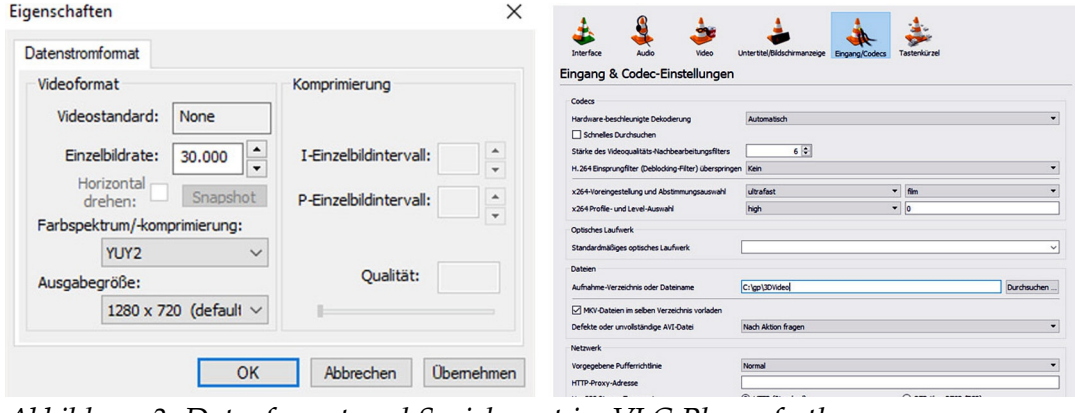

Abbildung 3: Datenformat und Speicherort im VLC Player festlegen

Verfahren Sie mit einer zweiten Instanz der Software auf die gleiche Weise. Starten Sie die Aufnahmen durch Klick auf den roten Button. Sie erhalten zwei Videoclips mit der Bezeichnung VLC-Record-*Datum-Uhrzeit*.avi im angegebenen Speicherpfad. Zur Synchronisation der Aufnahmen wurde die bekannte Hollywoodklappe am Anfang der Bildsequenz mit aufgenommen. Die gleiche Vorgehensweise steht Ihnen auch mit VirtualDub zur Verfügung.

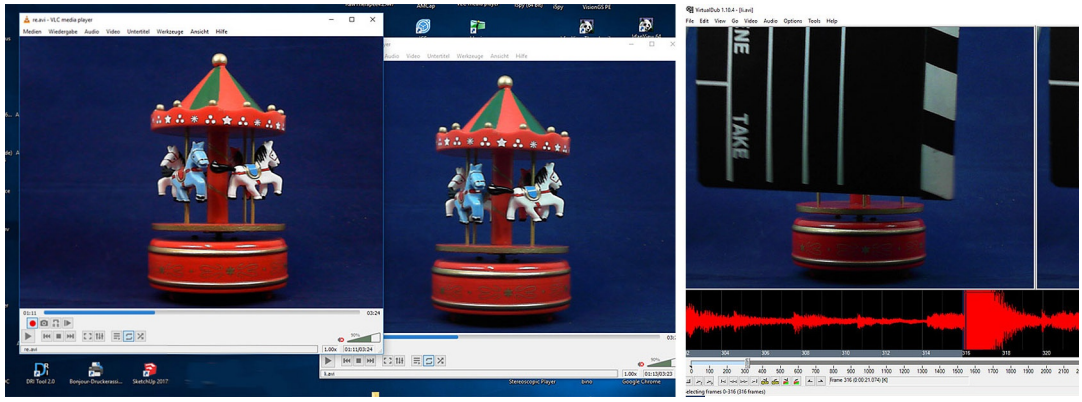

Abbildung 4: Zwei Instanzen von VLC laufen parallel. Rechts die Audiospur in VirtualDub

Webcams unter Linux

Unter Linux ist fswebcam ein einfaches Werkzeug zum Zugriff auf die Webcam. Der hier eingesetzte Rechner ist ein Raspberry Pi 3, der bei der 3D-Videoaufnahme doch arg gefordert ist. Zunächst schauen wir uns aber mal die Handhabung an. Einzelbilder einer Webcam erhält man mit dem Werkzeug fswebcam über die Eingabe einer Kommandozeile. Zur Überprüfung der Funktionsfähigkeit und Abgleichung der Kameraparameter sollte man dieses Tool einsetzen. Installieren Sie

```
sudo apt-get install fswebcam
```

und verbinden Sie die USB-Kameras. Über den Befehl

```
ls /dev/video*
```

bekommt man die erkannten Webcams als video0 und video1 angezeigt.

Die Abfrage der aktuellen Kameraeinstellungen erfolgt mit der Option -list-controls

```
fswebcam -d /dev/video0 --ls list-controls
```

alternativ auch durch

```
v4l2-ctl -d /dev/video0 -list-controls
```

Auf dem Monitor wird folgendes Ergebnis angezeigt:

```
Video input set to 0 (Camera 1: Camera, ok)
brightness (int)     : min=30 max=255 step=1 default=-8193 value=142
contrast (int)       : min=0  max=10  step=1 default=57343 value=10
saturation (int)     : min=0  max=200 step=1 default=57343 value=100
white_balance_temperature_auto (bool)    : default=1 value=1
power_line_frequency (menu)     : min=0 max=2 default=2 value=2
white_balance_temperature (int): min=2500 max=10000 step=1 default=57343
                                 value=3650 flags=inactive
sharpness (int)              : min=0 max=50 step=1 default=57343 value=25
backlight_compensation (int): min=0 max=10 step=1 default=57343 value=0
exposure_auto (menu)         : min=0 max=3 default=0 value=3
exposure_absolute (int)      : min=1 max=10000 step=1 default=156 value=156
```

```
                                    flags=inactive
pan_absolute (int)    : min=-529200 max=529200 step=3600 default=0 value=0
tilt_absolute (int)   : min=-432000 max=432000 step=3600 default=0 value=0
focus_absolute (int)  : min=0 max=40 step=1 default=57343 value=20
focus_auto (bool)     : default=1 value=0
zoom_absolute (int)   : min=0 max=317 step=1 default=57343 value=0
```

Der Benutzer hat Zugrifff auf die Werte und kann, ebenfalls über die Kommandozeile, mit der Option -s bzw. -c <Schlüsselwort> = <Wert> seine eigenen Festlegungen treffen.

Somit sind bei statischen Objekten auch programmgesteuert Belichtungsreihen durchzuführen. Setzt man beispielsweise *focus_auto=False*, so kann man anschließend die Werte um den Bereich von 25 für eine Fokusreihe im Nahbereich bei etwa 10 cm einsetzen. Mit der Abbildung 5 liegt ein Beispiel vor, das aus 7 Aufnahmen 'gestackt' wurde.

Abbildung 5: Blechspielzeug der Firma Pay aufgenommen mit Fokusstacking

Gibt man in der Kommandozeile ein:

```
fswebcam -v -S 2 -F 1 -r '640x480' -d /dev/video0 snap0.jpg
```

wird das Bild unter *snap0.jpg* gespeichert. Mit *-S 2* werden die ersten zwei Bilder übersprungen, *-F1* steht für einen Frame, *-r* gibt die Aulösung an, *-d* verweist auf das Gerät. Der Verbose-Mode *-v* erzeugt eine detaillierte Ausgabe des Capture-Vorgangs.

Die Speicherung von Einzelbildserien bewirkt man aber besser über ein Bash-Script, in dem die Dateinamen der Einzelbilder unter Benutzung von Zeit oder aufsteigendem Zähler generiert werden. Den Aufruf kann man zeitlich auch über sog. Cron-Jobs steuern. Die einzelnen Frames fügt man in der Nachbearbeitung zu einem im Video zusammen. Weitere umfangreiche Informationen zum Betrieb einer

Webcam unter Linux findet man im Internet bei den weiter unten angegebenen Referenzen.

Das direkte Aufzeichnung eines Video wird durch den Befehl *avconv* vorgenommen:

```
avconv -f video4linux2 -r 30 -s 640x480 -t 5 -i /dev/video0
-y recCam0.avi
```

Das Format ist Video-for-Linux, mit 30 fps, der Auflösung 480p im 4:3 Seitenformat für 5 Sekunden. In einer zweiten Shell setzen wir den gleich Befehl für video1 ab. Die Synchronisation erfolgt in der Nachbearbeitung mit der schon oben benutzten Hollywoodklappe. Mit beiden Clips geht es dann in das Schnittprogramm zur Synchronisation auf dem Host PC. Die Daten werden auf den Windows PC mittels einer SSH Kommunikation übertragen. Es ist zu erkennen, dass die Kommandozeilenbefehle für viele Aufgaben der Videobearbeitung ein mächtiges Werkzeug darstellen.

Mehr Komfort mit OpenCV

Mit OpenCV, der quelloffenen Computer Vision Bibliothek, öffnet sich ein weiteres Spektrum der Videobearbeitung. Die Installation von OpenCV 3 auf dem Raspberry Pi benötigt etliche Eingaben in der Kommandozeile und auch erheblichen Zeitaufwand beim Kompilieren. Wir gehen an dieser Stelle von einer Python 3 Einbindung aus. Auf eine detailliert ausgearbeitete Anleitung von Adrain Roesebrock, die man schrittweise gut nachvollziehen kann, wird weiter unten verwiesen. Bei Windows ist die Installation einfacher. Sofern man Python 3.4 installiert hat, kommt man mit einem pip-Kommando zurecht, vgl. ebenfalls die Quelle in den Referenzen.

Zunächst betrachten wir das Script zur Einzelbildaufnahme, nicht in allen Einzelheiten, sondern nur mit den notwendigsten OpenCV-Aufrufen:

```
import cv2                              # Einbindung von OpenCV
cam0 = cv2.VideoCapture(0)              # Instanziierung der Kameras
cam1 = cv2.VideoCapture(1)
…                                       # Festlegungen
for i in range (0,n ):                  # Schleife für n Bilder
   ret, frameL = cam0.read()
   ret, frameR = cam1.read()
   txt='v4l2-ctl -d /dev/video0 -c focus_absolute=' +str(f)
   os.system(txt)                       # Änderung der Kameraeinstellungen

cv2.imshow('Cam0',frameL)               # Bildanzeige
cv2.waitKey(0)
#
cv2.imwrite('bild_1.jpg', frameL)     # Bild speichern
```

Unser Script importiert die OpenCV Bibliothek, die beiden Kamera-Objekte werden deklariert, gefolgt von einer Schleife mit Bildaufruf und Parameteränderung für das Stacking. Die Frames werden angezeigt und gespeichert. Man mag die komfortable Handhabung der Bibliothek erkennen, Abbildung 5 wurde mit diesem kleinen Programm gesteuert. Dieses und das folgende Script der Videaufzeichnung können Sie von der angebotenen Supportseite downloaden, vgl. Referenzen.

Zur Videoaufnahme benutzen wir einen objektorientierten Ansatz. Es besteht eine Klasse *CamVid()*, mit den Methoden *captureVid()* und *saveVid()*. Im Konstruktor werden die Kameraparameter eingestellt und unser Programm benötigt nur noch die Instanzen des Objects CamVid und muss darauf die Methoden *captureVid()*und *saveVid()* in einer Endlosschleife anwenden. Abgebrochen wird bei Tastatureingabe *q*. Sehr übersichtlich. Das Riesenrad der Abbildung 6,8,9 stellen einen Frame von Testvideaos dar.

```
def main():
        cam1 = CamVid('left','out1.avi')
        cam2 = CamVid('right','out2.avi')
        while(True):
           cam1.captureVi()
           cam1.saveVid()
           cam2.captureVid()
           cam2.saveVid()
           if cv2.waitKey(1) & 0xFF == ord('q'):
              break
```

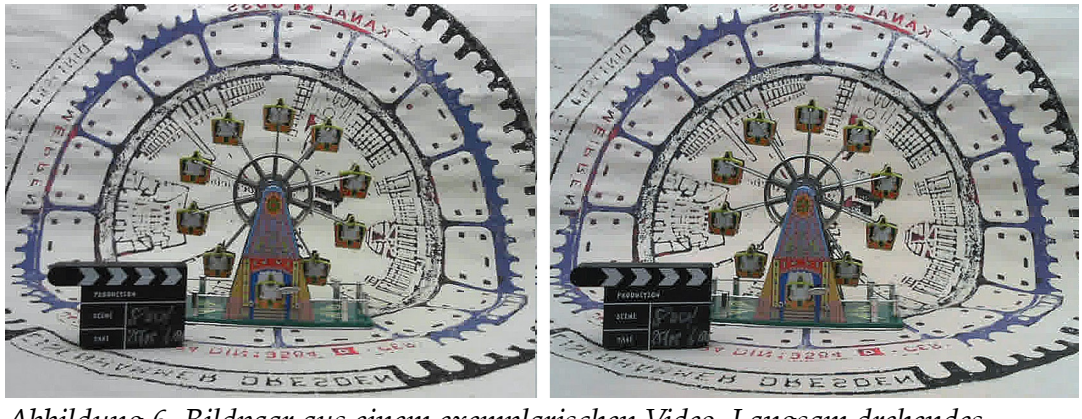

Abbildung 6. Bildpaar aus einem exemplarischen Video. Langsam drehendes Riesenrad

Visuelle und akustische Synchronisation mit VirtualDub

Den zeitgleichen Beginn der Videosequenzen kann man nicht nur akustisch mit der bewährten Hollywoodklappe markieren, wie es im rechten Teil der Abbildung 4 zu

sehen ist. Es geht auch visuell mit einem Blitzlicht. Beim Filmschnitt greifen wir zunächst zu Freeware VirtualDub und schneiden den Anfang bis zur Synchronisation raus. Bei Verfügbarkeit eines professionellen stereoskopischen Filmschnittprogramms kann natürlich auch dieses die Aufgabe erledigen.

Die visuelle Synchronisation mit dem Blitzlicht kann man in der Bildübersicht der Einzelbilder gut beobachten. Mit

```
ffmpeg -i input.avi links/frameL_%d.jpg
```

haben wir den Streifen in seine Einzelbilder zerlegt. Zwar ist der Blitz nicht zeilenweise exakt oder gar pixelgenau synchronisiert, tritt aber in beiden Streifen mit der Bildnummer 149 auf.

VirtualDub mag möglicherweise nicht den Codec der Bildaufnahme. Mit FFmpeg ist das ursprüngliche Videoformat aufzubreiten:

```
ffmpeg -i input.avi -vcodec rawvideo -pixfmt bgr24 output.avi
```

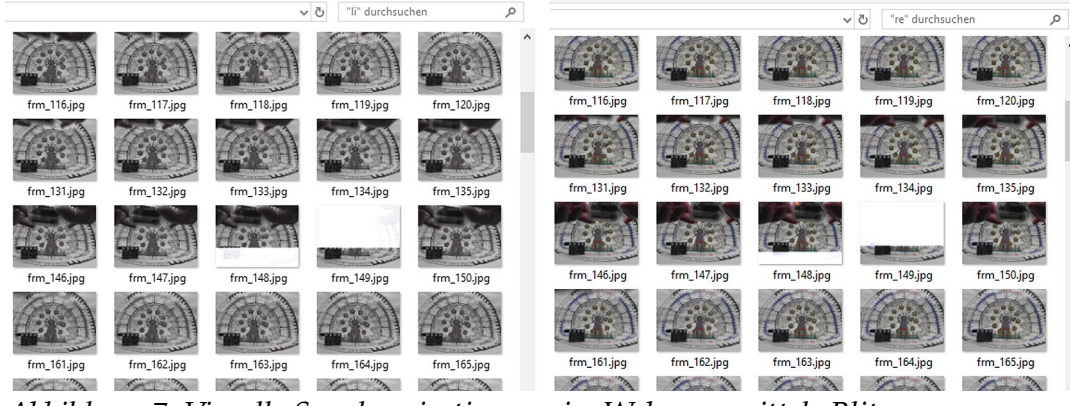

Abbildung 7: Visuelle Synchronisation zweier Webcams mittels Blitz

Die Montage der Bildsequenzen hinsichtlich Scheinfenster und die Feinausrichtung der vertikalen Parallaxe ist noch vorzunehmen. Sehr hilfreich ist hierbei der Stereo Movie Maker (SMM). Scheinfenster und Parallaxen sind mit SMM justierbar. Audiospuren werden im SMM leider nicht visualisiert. Mit einem stereoskopischen Player (Stereoscopic Player von Peter Wimmer oder Dino) ist man sehr flexibel in der Betrachtung der immer noch getrennt vorliegenden Videos. Möchte man ein Side-by-Side Format z.B. für YouTube erzeugen, nimmt man die Befehlszeile:

ffmpeg -i *li.avi* -i *re.avi* -filter_complex [0:v][1:v]hstack[v] -map [v] -metadata:s:v:0 stereo_mode=left_right -codec:v libvpx -aspect 8:3 -crf 10 *karussell.mkv*

Schauen Sie doch mal rein bei YouTube. Dort kann man auch die schnelle Bewegung des Kettenkarussels beobachten. Bei 1/30 Belichtungszeit wird die

Bewegung natürlich nicht eingefroren.

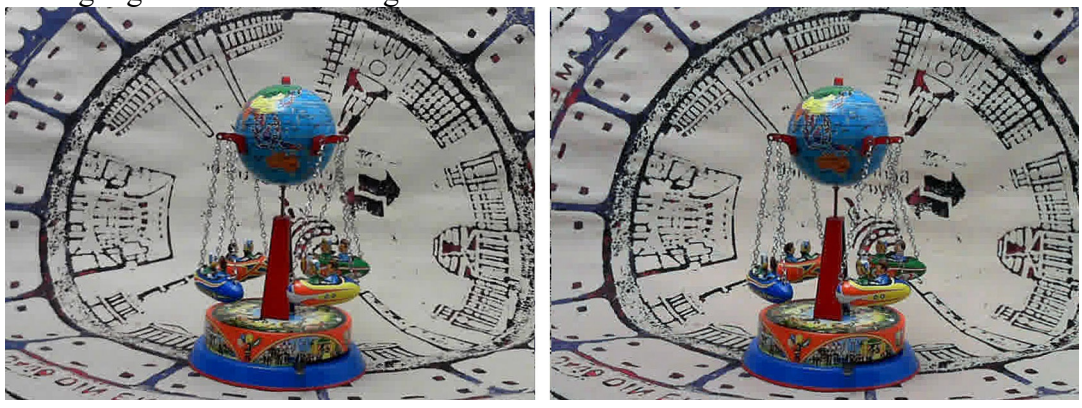

Abbildung 8: Kettenkarussel statisch

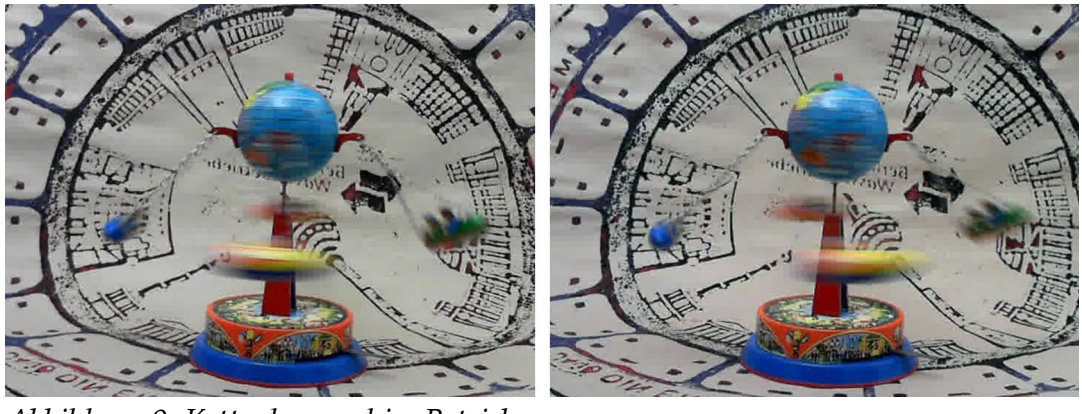

Abbildung 9: Kettenkarussel im Betrieb

Zusammenfassung

Mit minimaler Ausstattung und Freeware wird der Weg zum 3D-Raumbild bzw. 3D-Video mit zwei Webcams aufgezeigt. Selbstverständlich kann man nicht die Qualität professioneller Videokameras erreichen. Belichtet wird mit 1/30 s, das ist nicht die Anforderung an Hochgeschwindigkeit. Bei der Auflösung sollte man, gerade auch im Nahbereich, aber die Anforderungen nicht zu hoch ansetzen. Für gute Ergebnisse ist eine ausgewogene Beleuchtung Voraussetzung. Wenn auch VirtualDub und der Stereomovie Maker bereits etwas in die Jahre gekommen sind, so bieten die Werkzeuge in Kombination mit FFMpeg, OpenCV und VLC einen wirksamen Einstieg in die Videobearbeitung am PC unter Windows und Linux. Und gerade die Programmierung mit Zeitraffer, Bewegungserkennung u.a. sowie die Datendistribution über das Netz spielen eine Rolle bei der Anwendung von Webcams.

Referenzen

Installation von Python/OpenCV:

https://www.scivision.co/install-opencv-python-windows/
https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on- your-raspberry-pi/

Webcams unter Linux:

www.netzmafia.de/skripten/hardware/Webcam/
www.m-wulff.de/linux/raspberry/webcam/

Download der Python Scrpte:

www.3d.imagefact.de/support/webcamVideo.zip

Videomaterial 3D auf YouTube:

Suchen Sie auf YouTube.com den Kanal des Autors. Dort finden Sie einige experimentelle Video-Schnipsel in 3D