

Digitale AV-Technik - Aufgabenblatt 07

Vorbereitung

Verwenden Sie ihr eigenes Smartphone; auf dem Smartphone soll im Browser ein Video abgespielt werden. Das Video soll vorher auf dem PC vorverarbeitet und codiert werden.

Installation FFmpeg

Laden Sie [FFmpeg](#) für ihr Betriebssystem herunter und installieren Sie es auf Ihrem Laptop.

Prüfen Sie die Installation indem Sie auf der Konsole die folgenden Befehle eingeben:

```
ffmpeg -version
```

Es sollte die Version von FFmpeg ausgegeben werden. Dieses Aufgabenblatt wurde mit der Version 7.1 auf MacOS getestet.

```
ffprobe -version
```

Damit überprüfen Sie ob auch [ffprobe](#) mit installiert wurde. Ein Tool um den Codec und Container einer Datei oder eines Streams zu überprüfen.

Download der Testvideos

Klonen Sie sich das GitHub Repository mit den Python Skripten zum Kurs:

https://github.com/uhahne/DAVT_python

Folgen Sie der Installationsanleitung und laden Sie sich verschiedene Testvideos herunter. Unbedingt erforderlich sind die im Repository vorbereiteten Videos:

- [ITU-R Rec. BT.802, 525-line](#)
 - Download möglich bei [CVDL](#), beachten Sie, dass eine Registrierung notwendig ist.
- [StarCraft](#) von Gaelle Martin-Cocher. Die StarCraft Sequenz, die von Derf-Twitch zur Verfügung gestellt wurde, kann auch [hier](#) im y4m Format heruntergeladen werden.

Ziel

Komprimieren Sie die beiden Rohvideos mithilfe von [FFmpeg](#), sodass sie auf Ihrem eigenen Smartphone flüssig abgespielt werden kann. Finden Sie die optimale Kombination aus Codec, Auflösung und Bitrate, die mit dem Standardbrowser Ihres Geräts kompatibel ist.

Vorgehen

1. Ausgangssituation

Sie haben ein Rohvideo (z. B. im Format `.yuv` oder `.avi`). Ihr Ziel ist es, dieses Video so zu komprimieren, dass es:

- eine möglichst geringe Dateigröße hat,
- dabei eine hohe Videoqualität beibehält,
- und auf Ihrem Smartphone im Standardbrowser ohne Probleme abgespielt werden kann.

2. Recherche zur Codec-Kompatibilität

1. Öffnen Sie die Website CanIUse.com.

2. Finden Sie heraus, welche Video-Codecs vom Standardbrowser Ihres Smartphones unterstützt werden.

- **Tipp für Android:** Prüfen Sie die Kompatibilität mit Chrome.
- **Tipp für iOS:** Prüfen Sie die Kompatibilität mit Safari.

3. Recherchieren Sie die Unterstützung der folgenden Codecs:

- **H.264 (AVC)**
- **H.265 (HEVC)**
- **VP9**
- **AV1**

4. Notieren Sie die unterstützten Codecs und die jeweiligen Einschränkungen mit Hilfe der [Informationen von Mozilla](https://caniuse.com/) und erstellen Sie sich eine Liste der Codecs, die für ihr Smartphone funktionieren sollten.

3. Video-Komprimierung mit FFmpeg

Verwenden Sie FFmpeg, um die Videos zu komprimieren und vergleichen Sie die Resultate mit dem Python Skript `compare_video_player.py`.

Raw Video (.yuv)

Die Dateiendung `.yuv` bedeutet, dass es sich um ein Video im Rohformat handelt. Dieses können Sie mit `ffmpeg` in ein unterstütztes Format übertragen. Es lässt sich nicht direkt im Python Skript `compare_video_player` verwenden. Bei der Konvertierung in ein unterstütztes Format (z. B. `.mp4`) mit `ffmpeg` müssen Sie die Auflösung und das Pixelformat angeben. Beispiel:

```
ffmpeg -f rawvideo -vcodec rawvideo -s 1920x1080 -r 25 -pix_fmt yuv420p  
-i inputfile.yuv -c:v libx264 -preset ultrafast -qp 0 output.mp4
```

Experimentieren Sie mit den folgenden Parametern:

Codec

Testen Sie verschiedene Codecs (z. B. H.264 und H.265), die kompatibel mit Ihrem Browser sind.

Beispiel:

```
ffmpeg -i input.avi -c:v libx264 output.mp4
```

Falls Sie mit dem AV1 Codec experimentieren wollen, beachten Sie, dass `libaom-av1` sehr langsam ist, aber `libsvtav1` in der Regel schneller ist.

Auflösung

Passen Sie die Auflösung an die Eigenschaften Ihres Smartphones an (z. B. 1080p, 720p). Beispiel für 720p:

```
ffmpeg -i input.avi -vf scale=1280:720 -c:v libx264 output_720p.mp4
```

Bitrate

Finden Sie eine geeignete Bitrate, die eine gute Balance zwischen Qualität und Dateigröße bietet.

Beispiel:

```
ffmpeg -i input.avi -b:v 2M -c:v libx264 output_2mbps.mp4
```

4. Überprüfung und Optimierung

1. Übertragen Sie die komprimierte Videodatei auf Ihr Smartphone.
 2. Öffnen Sie das Video im Standardbrowser und überprüfen Sie:
 - Spielt das Video flüssig ab?
 - Gibt es Qualitätsprobleme (z. B. Artefakte)?
 - Ist die Datei klein genug für eine schnelle Ladezeit?
 3. Falls nötig, passen Sie die Einstellungen in FFmpeg an (z. B. Codec, Bitrate oder Auflösung).
 4. Speichern Sie die komprimierten Videos und fügen Sie jeweils eine kurze Dokumentation hinzu, die Folgendes beschreibt:
 - Den verwendeten Codec und warum Sie ihn gewählt haben.
 - Die Auflösung und Bitrate des Videos.
 - Eine kurze Einschätzung der Qualität des Videos und die Unterschiede zum Ausgangsvideo.
 - Hierzu können Sie das Skript `compare_video_player.py` aus dem [GitHub Repository](#) verwenden.
 - Herausforderungen, die Sie bei der Komprimierung und Wiedergabe gelöst haben.
 - Angaben zum verwendeten Smartphone-Modell und Browser.
-