

Aufgaben zu Onlinekodierung und -übertragung

Lösen Sie die nachfolgenden Aufgaben und bereiten Sie diese bis zum nächsten Lehrveranstaltungstermin vor. Unterstrichene Aufgaben sind nach Möglichkeit während der Lehrveranstaltung zu lösen.

LB-OKÜ 01.

- a) Senden Sie die im MP4-Format zur Verfügung gestellte Testsequenz mit *FFmpeg* MPEG-TS-gekapselt über RTP an eine ausschließlich lokale frei verwendbare Multicastadresse. Transkodieren Sie den Videostrom zu H.264 mit einem Zehntel der Originalauflösung und einer Ihnen vernünftig erscheinenden Bitrate. Verwenden Sie zur Skalierung den Ausgabeparameter `-vf "scale=BxH"`, wobei B und H die gewünschte Breite bzw. Höhe in Pixeln sind. Verifizieren Sie die ausgesandten Daten durch Abspielen derselben auf einem anderen Rechner mit *ffplay* und der URL als Parameter. *Hinweis: Einige von x264 bekannten Parameter, z.B. -preset ultrafast können verwendet werden, um die Videokodierung zu beschleunigen.*
- b) Modifizieren Sie Ihre Einstellungen aus a) derart, dass Sie eine höhere Videoauflösung (maximal die der Originalsequenz) und eine jeweils dazu passende Bitrate verwenden. Bestimmen Sie jene Auflösung, bis zu der das Video unterbrechungsfrei gestreamt werden kann und begründen Sie die Unterbrechungen bei höheren Auflösungen durch geeignete vom Betriebssystem zur Verfügung gestellte Messwerte des Sende- und/oder Empfängers. Verwenden Sie zusätzlich die Fehler- und Statusmeldungen von *FFmpeg* auf der Konsole beim Sender und Empfänger.

LB-OKÜ 02.

- a) Kodieren Sie die zur Verfügung gestellte Testsequenz sowie jene aus LB-VK 01. mit der modifizierten Version von *NAL-SIM* (verfügbar im E-Learning-System, Aufruf: `wine NAL-SIM.exe`) mit einer Bitrate für mittlere Qualität und simulieren Sie anschließend einen Übertragungskanal mit einer Verlustrate von 5% und einer *Avg. burst length* von 1. Bewerten Sie die in den dekodierten Sequenzen auftretenden Bildfehler in Hinblick auf Häufigkeit und Schwere und nutzen Sie den *Stream Analyzer* rechts oben in *NAL-SIM*, um verloren gegangene NAL-Units und deren Typen ausfindig zu machen. *Hinweis: Verwenden Sie keine Dateipfade (Felder vgl. Anleitung) in gemeinsamen Ordnern, sondern in C:\ (physisch unter ~/.wine/drive_c).*
- b) Wiederholen Sie die Simulationen aus a) mit Verlustraten von 1 und 10%. Erläutern Sie den Einfluss der Verlustrate auf die Bildfehler. *Hinweis: Berücksichtigen Sie bei den Ergebnissen, dass das Verwerfen probabilistisch geschieht und daher bei mehreren aufeinander folgenden Durchgängen nicht exakt wiederholbar ist.*

LB-OKÜ 03.

- a) Wiederholen Sie LB-OKÜ 02. a) mit vier Slices pro Frame. Erläutern Sie die Unterschiede zu den Ergebnissen aus LB-OKÜ 02. a).
- b) Wiederholen Sie LB-OKÜ 02. a) mit aktiviertem FMO (Typ 1) und zwei Slice groups. Erläutern Sie die Unterschiede analog zu a).
- c) Analysieren Sie den Einfluss der Anzahl der ME-Referenzbilder auf die Ergebnisse aus LB-OKÜ 02. a).

LB-OKÜ 04.

- a) Schreiben Sie auf Basis von LB-MTM 04. a) ein Programm, das den Kodierungsdelay jedes Videoframes misst und in Millisekunden auf der Konsole ausgibt. Nutzen Sie das `opaque`-Feld von `x264_picture_t`, um jedem Videoframe einen Zeitstempel der aktuellen Uhrzeit zuzuordnen. Dieses Feld wird von der `libx264` durchgereicht und kann aus dem dazugehörigen kodierten Frame wieder ausgelesen werden. Aus der Differenz zwischen der aktuellen Uhrzeit nach der Kodierung und dem ausgelesenen ursprünglichen Zeitstempel kann der Kodierungsdelay ermittelt werden. Nutzen Sie zur Ermittlung der aktuellen Uhrzeit die Funktion `clock_gettime` mit dem Parameter `CLOCK_MONOTONIC` (vgl. Dokumentation unter <https://en.cppreference.com/w/c/chrono/clock>).
- b) Passen Sie Ihr Programm aus a) derart an, dass anstatt der Kodierungsdelays der einzelnen Frames eine Statistik ausgegeben wird, die die durchschnittlichen und maximalen Kodierungsdelays in Millisekunden ausgibt. Die Kodierungsdelays sollen dabei pro Frametyp (I, P, B) aufgeschlüsselt werden. Verwenden Sie zur Frametypbestimmung die `libx264`-Makros `IS_X264_TYPE_I`, `IS_X264_TYPE_P` und `IS_X264_TYPE_B`.
- c) Passen Sie Ihr Programm aus b) derart an, dass der maximale Kodierungsdelay für ein 30-fps-Video entweder einer Framedauer entspricht oder im Vergleich zu b) zumindest halbiert wird.