

Online-Kodierungstechniken am Beispiel H.264

Medientechnologie IL

Andreas Unterweger

Vertiefung Medieninformatik und Bildverarbeitung
Studiengang ITS
FH Salzburg

Sommersemester 2023

- Onlinekodierung
 - Kodierung (und eventuell Dekodierung) in Echtzeit
 - Laufender Dateninput (Verzögerungen unerwünscht)
 - Eventuell fehlerbehaftete Übertragung
- Ausgewählte Online-Kodierungstechniken
 - Echtzeitkodierung
 - Delay-Minimierung
 - Fehlerminimierung
- Nicht behandelte Aspekte (Auswahl)
 - Hardwareoptimierung
 - Softwareoptimierung
 - Netzwerkoptimierung
 - Übertragungsoptimierung
 - Skalierbarkeit
 - Qualitätssicherung

- Fernseh-Live-Übertragungen (z.B. Fußball-WM-Spiel)
 - Niedriger Delay
 - Übertragungsfehler zum Teil tolerierbar
- Video on Demand
 - Hoher Delay teilweise akzeptabel
 - Übertragungsfehler unerwünscht, aber tolerierbar
- Videokonferenz
 - Niedrigstmöglicher Delay
 - Übertragungsfehler zum Teil tolerierbar
- Telemedizin (z.B. Fernoperationen)
 - Niedrigstmöglicher Delay notwendig
 - Übertragungsfehler inakzeptabel

→ Verschiedene Anforderungen je nach Anwendungsgebiet

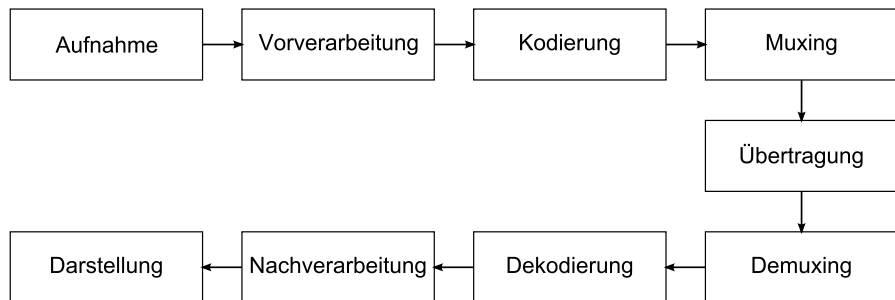
Definition: Echtzeit

- Echtzeitberechnung: Berechnungen haben Zeitvorgabe
- Deadline pro Berechnung (in Absolut- oder Relativzeit)
- Klassifizierung nach Folgen bei Deadlineüberschreitung
 - Harte Echtzeit: Fataler Fehler
 - Feste Echtzeit: Ergebnis ist unbrauchbar
 - Weiche Echtzeit: Qualitätsverlust wächst mit Überschreitung
- Kontext Online-Videokodierung (bei n fps)
 - Kodierung von n Frames pro Sekunde
 - Entgegennahme eines Frames alle $\frac{1}{n}$ Sekunden
 - Feste Echtzeit
- Kontext Videodekodierung (bei n fps)
 - Dekodierung von n Frames pro Sekunde
 - Ausgabe eines Frames alle $\frac{1}{n}$ Sekunden
 - Weiche Echtzeit

- Kodierparameter beeinflussen Encoder-Geschwindigkeit
- Haupteinflussfaktoren (zumeist unveränderbar)
 - Bildauflösung
 - Bildwiederholrate
 - Subsampling
- Haupteinflussfaktoren (veränderbar)
 - Anzahl auszuprobierender Modi/Partitionierungen
 - Anzahl Referenzbilder
 - Bewegungssuchradius
 - Aktivierter Loop-Deblocking-Filter
 - Entropiekodierungsalgorithmus
 - Bitrate
- Nebeneinflussfaktoren (veränderbar)
 - Bewegungssuchmetrik
 - I-Frame-Frequenz
 - ...

- Verzögerung (englisch *delay*) in der Datenverarbeitung und -übertragung (akkumuliert)
- Arten von Delay im Kontext der Video(-de-)kodierung
 - Kodierungsdelay (englisch *encoding delay*)
 - Dekodierungsdelay (englisch *decoding delay*)
- Weitere Arten von Delay im erweiterten Kontext
 - Aufnahmedelay (englisch *capture delay*)
 - Vor-/Nachverarbeitungsdelay (englisch *pre- bzw. post-processing delay*)
 - Muxing-/Demuxing-Delay
 - Übertragungsdelay (englisch *transmission delay*)
 - Darstellungsdelay (englisch *display delay*)
 - ...

Übersicht Delay II



- Summe der Einzeldelays ergibt Gesamtdelay
- Fokus: Kodierung und Dekodierung
- Ziel: Delayminimierung

- Kodiergeschwindigkeit entspricht Kodierungsdelay
- Kodiergeschwindigkeit hängt von Kodierparametern ab (vgl. Kodierparameter auf Folie 5)
- Pufferung von Eingangsbildern
 - Ermöglicht „Vorkodierung“ (beeinflusst Echtzeitfähigkeit positiv)
 - Ermöglicht optionale Analyse zur Bitratensteuerung
 - Erhöht Delay proportional zu Puffergröße in Frames
 - Notwendig bei Verwendung von B-Frames
- B-Frame-Delay
 - Kodierreihenfolge weicht von Darstellungsreihenfolge ab
 - Delay: Maximale Differenz zwischen Kodier- und Darstellungsreihenfolge (Delay proportional zu maximaler Differenz)
 - Durch Verzicht auf B-Frames eliminierbar (Effizienzeinbußen)

- **Ausgabedelay**
 - Bereits kodierte Makroblöcke müssen in NALUs gekapselt werden
 - NALU enthält im Regelfall eine Slice
 - Höhere Anzahl Slices verringert Anzahl Makroblöcke pro Slice
 - Fertige NALUs können schneller übertragen werden
- **Parallelisierung (bei Mehrkern-CPU's)**
 - Keine Prädiktion über Slice Grenzen
 - Unabhängige Verarbeitung von Slices möglich
 - Höhere Anzahl Slices ermöglicht Parallelverarbeitung
- **Nachteile durch Erhöhung der Sliceanzahl**
 - Verringerung der Kodiereffizienz (mehr Prädiktionsgrenzen)
 - Overhead durch Slice- und NALU-Header
 - Mögliche Erhöhung der Kodierdauer

- Dekodiergeschwindigkeit entspricht Dekodierungsdelay
- Dekodiergeschwindigkeit hängt von Kodierparametern ab
 - Einzelbildauflösung
 - Keine bzw. große Partitionen vereinfachen Dekodierung
 - Weighted Prediction und B-Blöcke erfordern zusätzliche Gewichtung
 - Geringe Datenrate reduziert Entropiedekodierungsaufwand
 - Verzicht auf Loop-Deblocking-Filter spart Rechenoperationen
 - Slices ermöglichen parallele Dekodierung (bei Mehrkern-CPU's)
 - ...
- B-Frame-Delay verzögert Ausgabe
- Niedrige I-Frame-Frequenz erhöht Random-Access-Dauer beim ersten Dekodierungsvorgang (z.B. Ein- oder Umschalten)

- Ungültige Videodaten
 - Falsche Koeffizienten(-vorzeichen)
 - Falsche signalisierte Modi
 - Falsche signalisierte Partitionen
 - Falsche Bewegungsvektoren
 - Fehlerfortpflanzung durch Intra- und Interprädiktion→ Schwere Bildfehler
- Ungültige entropiekodierte Daten
 - Daten bis zu nächstem Resynchronisationspunkt unlesbar→ Fehlende Bildteile
- Ungültige Steuerdaten
 - Ungültige SPS/PPS→ Video (eventuell) nicht dekodierbar
- Fehlende Daten: Bilder oder Bildteile fehlen (Erkennung?)

Folgen von Übertragungsfehlern: Beispiel

- Verlust eines schmalen I-Slices (eine Makroblockzeile)
→ Naiver Interpolationsversuch missglückt



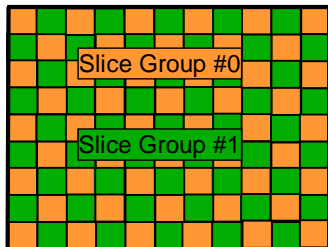
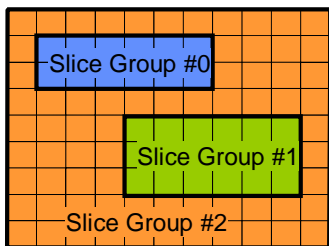
Quelle: Boulos, F., Chen, W., Parrein, B. und Le Callet, P.: A new H.264/AVC error resilience model based on Regions of Interest. 17th International Packet Video Workshop, 2009, pp. 1-9, 2009.

- Anzahl I-Frames bzw. I-Blöcke erhöhen
 - Verringert zeitliche Fehlerfortpflanzung
 - Hoher Overhead (Kodiereffizienzverlust)
- Anzahl Slices erhöhen
 - Beschränkt örtliche Fehlerfortpflanzung
 - Moderater Overhead
 - Nur wirkungsvoll, wenn Slices in getrennten Paketen übertragen werden
→ Hoher Übertragungsprotokolloverhead
- SPS und PPS in regelmäßigen Abständen wiederholen
 - Erlaubt Wiederaufnahme von Dekodierung
 - Geringer Overhead
 - Praktisch in Verwendung

- Slices mehrfach senden
 - Wichtige Slices können öfter gesendet werden
 - Exakte Datenduplikate
 - Hoher Overhead
- Decoder kann auf Datenverlust reagieren
 - Duplikate ersetzen verloren gegangene Slices
 - Duplikate ersetzen bei Bedarf fehlerhafte Slices
 - Kopien erfolgreich dekodierter Slices werden ignoriert
- Praktische Vorteile
 - Relativ einfach zu implementieren
 - Encoder kann über Wichtigkeit der Daten entscheiden
- Praktische Nachteile
 - Von keinem Mainstream-Encoder unterstützt
 - Bereitet Schwierigkeiten in Protokollen mit Zeitstempeln

Flexible Macroblock Ordering (FMO) I

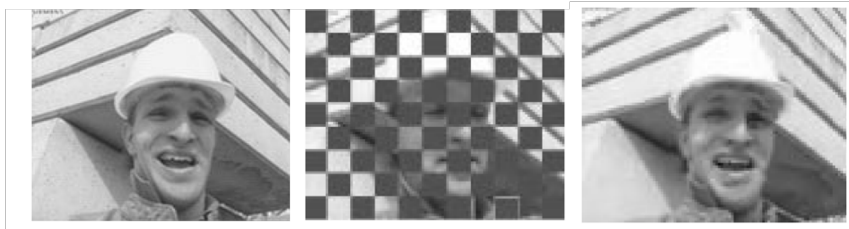
- Makroblockübertragungsreihenfolge weicht von Standard ab
 - Wichtige Bildregionen können im Datenstrom verteilt werden
 - Geringere Anfälligkeit auf gehäufte (englisch *burst*) Übertragungsfehler
- Werkzeug: Slice groups (können mehrere Slices enthalten)



Quelle: Wiegand, T. und Sullivan, G. J.: The H.264 | MPEG-4 AVC Video Coding Standard.
http://ip.hhi.de/imagecom_G1/assets/pdfs/H264_03.pdf (14.10.2010), 2004.

Flexible Makroblock Ordering (FMO) II

- Verschiedene Variationen möglich
- Kodierung aufwändig
 - Anpassung der Kodierungs- und Übertragungsreihenfolge
 - Berücksichtigung von Nachbarblockverfügbarkeit bei Intraprediktion
- Schachbrettmuster erlaubt Rekonstruktion aus vorherigem Frame und aktuellem Frameteil (Rekonstruktion nicht standardisiert)



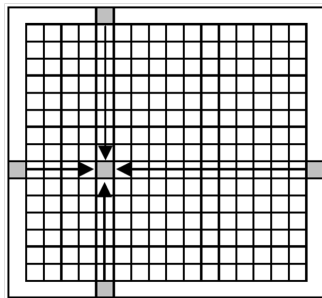
Quelle: Kolkeri, V. S., Koul, M. S., Lee, J. H. und Rao, K. R.: Error Concealment Techniques in H.264/AVC For Wireless Video Transmission In Mobile Networks.

http://students.uta.edu/ms/msk3794/docs/ERROR_CONCEALMENT_TECHNIQUES-JAES2009.pdf (14.10.2011), 2008.

Auswahl einfacher Fehlerverschleierungsmethoden I

- Gewichtete Mittelung (englisch *weighted averaging*): Interpolation aus verfügbaren Nachbarn (kleinere Gewichte für weiter entfernte Nachbarn, größere für näher liegende)

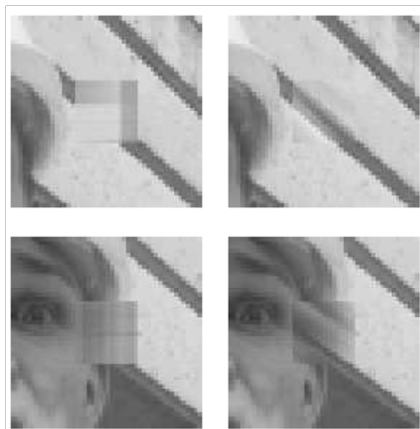
→ Schlechter, wenn Nachbarn ebenfalls interpoliert



Quelle: Kumar, S., Xu, L., Mandal, M. K. und Panchanathan, S.: Error Resiliency Schemes in H.264/AVC Standard. Elsevier Journal of Visual Communication & Image Representation (Special issue on Emerging H.264/AVC Video Coding Standard), Vol. 17(2), 2006.

Auswahl einfacher Fehlerverschleierungsmethoden II

- Interpolationsrichtung durch Kantenerkennung (rechts)
- Gewichtete Mittelung (links) zum Vergleich



Quelle: Nemethova, O., Al-Moghrabi, A. und Rupp, M.: Flexible Error Concealment for H.264 Based on Directional Interpolation. 2005 International Conference on Wireless Networks, Communications and Mobile Computing, vol.2, pp. 1255-1260, 2005.

Auswahl einfacher Fehlerverschleierungsmethoden III

- Bewegungsvektoren bei erkanntem Szenenwechsel null setzen

Bild n - 1



Bild n



Original

Mit Fehler

Korrigiert

Quelle: Su, L., Zhang, Y., Gao, W., Huang, Q. und Lu, Y.: Improved error concealment algorithms based on H.264/AVC non-normative decoder. 2004 IEEE International Conference on Multimedia and Expo, vol.3, pp. 1671-1674, 2004.

Fragen?