**Opgave 1: x264**

1. 1. **Geef de gevraagde grafiek en geef je motivering voor de gebruikte rate points.**

De ratepoints gaan van 1 t.e.m. 51 met stappen van 5. Deze liggen dicht genoeg bij elkaar, zonder dat lange wachttijden optreden.

* 1. **Verklaar de gebruikte commandolijnparameters voor de gevraagde configuraties.**

|  |  |
| --- | --- |
| **Parameter** | **Uitleg** |
| --tune psnr | Encoder afstemmen voor optimale PSNR |
| --fps 25 | Het aantal frames per seconde |
| --frames 50 | Het aantal beelden om te encoderen |
| --input-res 416x240 | Resolutie van een frame |
| --keyint 1 | Het maximum aantal beelden tussen 2 I-frames |
| --min-keyint 1 | Het minimum aantal beelden tussen 2 I-frames |
| --b-adapt 0 | Aan- of uitzetten van de methode om optimale aantal B-frames te kiezen |
| --bframe 0 | Het maximum aantal B-frames na elkaar |

* 1. **Hoe verhouden de RD-curves zich tegenover elkaar? Wat kun je hieruit afleiden?**

De RD-curves hebben allen dezelfde vorm. Verschillende configuraties beïnvloeden de punten op de RD-curves op dezelfde manier.

* 1. **Kijk ook naar de afstand tussen de verschillende RD-curves. Wat valt hier op?**

De eerste configuratie is duidelijk slechter. Deze bevat meer bits per seconde voor eenzelfde PSNR bij een vaste kwaliteitsfactor q. De curves van de 2de en 3de configuratie liggen veel dichter op elkaar.

* 1. **Verklaar waarom dit gedrag optreedt.**

Config 1: Ieder I-frame (alle frames dus) hebben een POC van 0. Dit is nu eenmaal zo bij AVC encoding. De hop count wordt gereset bij elk I-frame. I-frames zijn onafhankelijk van elkaar.

Config 2: Eerst komt het I-frame met POC 0, daarna komen de 15 P-frames in volgorde. De P-frames zijn afhankelijk van voorgaande frames en moeten dus in de correcte volgorde staan.

Config 3: Eerst komt het I-frame met POC 0, daarna een P-frame met POC 10 en vervolgens 4 B-frames. B-frames hangen zowel af van voorgaande als toekomstige frames en kunnen dus door elkaar staan.

* 1. **In welke gevallen zou je in de praktijk wel nog kiezen voor de minst compressie-efficiënte configuratie?**

Op een tv kan het beeld enkel veranderen wanneer de volgende frame een I-beeld is. Met enkel I-beelden comprimeren is niet efficiënt, maar daarmee kan wel onmiddellijk van beeld versprongen worden, zonder dat je moet wachten tot een I-frame ingeladen is.

1. 1. **Stel voor de twee mogelijke combinaties een RD-curve op en combineer die in één grafiek in het verslag. Geef ook de twee gebruikte commando's.**

**Config 4:** x264.exe -q [quantisatieparameter] -v -o [output].264 --tune psnr --fps 25 --frames 50 --input-res 416x240 --keyint 16 --min-keyint 16 --bframe 4 --b-adapt 0 **--subme 0 --partitions none** [sequentie].yuv

**Config 5:** x264.exe -q [quantisatieparameter] -v -o [output].264 --tune psnr --fps 25 --frames 50 --input-res 416x240 --keyint 16 --min-keyint 16 --bframe 4 --b-adapt 0 **--subme 11 --partitions all** [sequentie].yuv

* 1. **Beschrijf de waargenomen invloed van de parameters op de codeerefficiëntie en de codeersnelheid en geef hiervoor een mogelijke verklaring.**

Hoe hoger de precisie van de bewegingsvectoren, hoe beter, maar hoe trager.

Hoe meer partities geanalyseerd worden, hoe hoger de kwaliteit, maar dit gaat ten koste van de snelheid.

Dit zien we ook op de grafiek.



**Config 6:** --me tesa --subme 11 --partitions all

**Config 7:** --me dia --subme 0 --partitions all

Logische keuze tussen laag en hoge nauwkeurigheid en oppervlakking en grondig zoekalgoritme.

* 1. Om een film op DVD op te slaan wil je een hoge complexiteit en een hoge compressierate, de uitvoeringstijd zal daardoor trager zijn. Om live beelden door te sturen heb je geen tijd om lang te encoderen. Hier kies je dus best voor een algoritme dat niet te veel tijd in beslag neemt.



We zien aan de RD-curves dat VBR optimaler encodeert dan CBR.

x264.exe -B B/1000 -v -o [output].264 --nal-hrd cbr --vbv-maxrate B/1000 --vbv-bufsize B/25000 --tune psnr --fps 25 --frames 50 --input-res 416x240 --keyint 16 --min-keyint 16 --bframe 0 --b-adapt 0 [sequentie].yuv



Bij CBR zijn alle frames uiteraard even groot. Voor VBR zijn de I-frames duidelijk groter. De P-frames worden vergeleken met de I-frames en worden zo gecomprimeerd.

De kwaliteit voor VBR is eerder constant, terwijl er bij CBR meer variatie is.

Door de constante bitrate moeten sommige frames meer gecomprimeerd worden, waardoor er een groot kwaliteitsverlies is.



VBR: 109 376 B / (1766 225 B/s /8) = 3963.903889 ms

Gemiddeld: 2560.394303 ms

CBR: 8 830 B /(220 744 B/s /8) = 39.994904 ms

Gemiddeld: 39.993817 ms

De duur om het grootste frame te versturen ligt ver van het gemiddelde, hierdoor wordt CBR voor live feed gekozen.



Videobellen: CBR

video on demand services zoals Netflix: CBR

video surveillance: CBR

Films opgeslagen op fysieke opslagmedia: VBR



De PSNR verkleint met elke iteratie. We verliezen dus kwaliteit.



In de limiet van de bitrate, benaderen we de PSNR van b0.

* 1. We kunnen concluderen dat hercoderen niet efficiënt is. We blijven kwaliteit verliezen, tenzij we de bitrate oneindig groot maken.

**Opgave 2: Alternatieve compressiealgoritmen en**

**invloed van de beeldinhoud**

1. We kozen voor –subme 4 aangezien –preset veryslow deze automatisch op 4 zet. We kozen in beide gevallen voor –me dia.

Er treedt betere compressie op bij video’s waarbij de opeenvolgende frames weinig van elkaar verschillen.

**Opgave 3: Kostenplaatje van videocompressie**