

# Verslag Video Compressie

Wout Coenen 1334370

Reinaert Van de Cruys 1334947

## Algemeen

We denken alle gevraagde functionaliteit correct geïmplementeerd te hebben. We hebben onze codec uitgebreid getest op de gegeven samples en op samples van <http://trace.eas.asu.edu/yuv/>, en we hebben geen artefacten kunnen vaststellen.

## Metingen

Voor de metingen beschouwen we de vijf gegeven samples, en een selectie van vier samples van de vernoemde website:

1. **Akiyo**, 300 frames, beeld van een nieuwslezeres waarbij enkel het gezicht beweegt.
2. **Bus**, 150 frames, beeld dat een bus volgt die door het verkeer rijdt.
3. **Foreman**, 300 frames, beeld van de man op het bouwterrein uit de les.
4. **Waterfall**, 260 frames, beeld van een waterval en bladerdak dat langzaam uitzoomt.

De resolutie van alle samples is 352x288, om de simpele reden dat we geen samples in yuv420 formaat hebben gevonden met een andere resolutie. We geloven niet dat de resolutie de kwaliteit of de compressie ratio merkbaar zou beïnvloeden, tenzij bij extreem lage resoluties. Verder gebruiken we voor alle metingen RLE, en we gebruiken volgende kwantisatie matrix:

2	4	8	16
4	4	8	16
8	8	32	64
16	32	64	128

We variëren de quantization matrix en RLE parameter niet, omdat het aantal testen anders gigantisch wordt, en we al inzicht hebben in hun effect vanwege de vorige opdracht. In plaats daarvan focussen we op hoe P-frames de compressie ratio van de voorgaande technieken nog kunnen verbeteren. Om die reden tonen we eerst de compressie ratio van alle samples met enkel I-frames (dus gop=1), als baseline.

## Compressie zonder P-frames (GOP size = 1)

00	01	02	03	04	akiyo	bus	foreman	waterfall
39.65%	28.17%	37.75%	17.83%	17.92%	31.66%	47.18%	39.24%	45.23%

De compressie ratio ligt tussen 1:2 (voor de waterval met veel detail) en 1:6 (voor de effen witte blokjes op effen zwarte achtergrond). Voor ons zijn dit verwachte resultaten, overeenkomstig met de ratio die we behaalden in de image compression assignment.

We kijken nu naar de compressie met (bijna) uitsluitend P-frames. We stellen de gop parameter in op 300, waardoor de codec één I-frame opslaat en daarna enkel P-frames. We doen dit voor verschillende merange waarden, waarbij we bewust niet simpelweg exponenten van twee nemen, maar juist zoveel mogelijk variëren. Onze codec slaat de motion vectors op met zo weinig mogelijk bits, dus bij hogere merange waarden nemen de vectoren meer ruimte in beslag.

## Compressie met (bijna) enkel P-frames (GOP size = 300)

	merange				
	1	2	7	20	50
<b>sample00</b>	31.18%	30.96%	30.83%	30.76%	30.71%
<b>sample01</b>	29.11%	28.31%	26.42%	25.62%	25.61%
<b>sample02</b>	25.30%	25.22%	24.88%	24.89%	24.91%
<b>sample03</b>	18.18%	18.10%	17.85%	17.85%	17.85%
<b>sample04</b>	18.33%	18.19%	18.02%	17.88%	17.88%
<b>akiyo</b>	23.30%	23.34%	23.47%	23.53%	23.54%
<b>bus</b>	49.87%	48.93%	41.33%	41.21%	41.12%
<b>foreman</b>	36.19%	35.50%	34.28%	33.58%	33.69%
<b>waterfall</b>	35.60%	35.61%	35.64%	35.77%	35.87%

Onze eerste vaststelling is dat niet alle samples baat hebben gehad bij het gebruik van P-frames. De twee samples met het bewegende witte vierkant (sample03 en sample04) halen nog quasi dezelfde compressie ratio (verschil <0.5%). Dit komt omdat ze zo weinig informatie bevatten, en onze image codec ze al zo efficiënt kon comprimeren dat er niets te verbeteren viel. De andere samples halen uiteenlopende resultaten, afhankelijk van hoeveel beweging ze bevatten. Stille beelden als Akiyo maken een sprong van 31.66% naar 23.30%, terwijl dynamische beelden zoals die van de vliegende draakjes (sample01) van 28.17% naar hooguit 25.61% gaan.

Onze tweede vaststelling is, zoals verwacht, dat niet alle samples baat hebben bij hogere meranges. Bij stille beelden als Akiyo zorgt een hogere merange juist voor een verslechtering, doordat de vectoren meer plaats zullen innemen. In andere samples zien we dat P-frames pas echt effect beginnen te hebben vanaf een bepaalde merange waarde, die ongetwijfeld rechtstreeks verband houdt met de snelheid waarmee objecten door het beeld bewegen. In de Bus sample bijvoorbeeld zien we een verslechtering in compressie bij merange waarden 1 en 2, maar springen we plots van 48.93% naar 41.33% bij merange waarde 7. Nadien halen we nog nauwelijks 0.2% verbetering, zelfs bij merange waarde 50, wat erop wijst dat de meeste objecten door het beeld bewegen aan een snelheid tussen 2 en 7 pixels per frame.

Als laatste kijken we naar de compressie voor verschillende GOP sizes, met een merange van 7 (waarvan we inmiddels weten dat hij voor alle samples een goed resultaat oplevert). Zoals verwacht krijgen we voor de samples die baat hebben bij P-frames kleine toenames in compressie naarmate de GOP size toeneemt.

## Compressie voor verschillende GOP sizes (merange = 7)

	gop				
	2	3	4	7	13
<b>sample00</b>	35.08%	35.52%	32.75%	31.84%	31.39%
<b>sample01</b>	26.94%	26.68%	26.42%	26.38%	26.24%
<b>sample02</b>	30.85%	28.70%	27.65%	26.45%	25.58%
<b>sample03</b>	17.84%	17.84%	17.85%	17.86%	17.86%
<b>sample04</b>	17.94%	17.96%	17.97%	17.98%	17.99%
<b>akiyo</b>	27.43%	26.09%	25.42%	24.56%	24.06%
<b>bus</b>	43.91%	42.85%	42.44%	41.84%	41.50%
<b>foreman</b>	36.35%	35.52%	35.08%	34.64%	34.45%
<b>waterfall</b>	40.24%	38.60%	37.81%	36.83%	36.17%

We zien dat er een goede reden is om regelmatig terug een I-frame op te slaan. Met één lange reeks P-frames (GOP size = 300) krijgen we een soort staart-effect te zien achter bewegende objecten. Dit zijn kleine fouten in de error correction, omdat die lossy wordt opgeslagen, en het neemt na verloop van tijd af, waardoor het als een staart achter het object lijkt te hangen. Door regelmatige herhaling van I-frames echter wordt dit effect onmerkbaar. We moeten de GOP size instellen op 10 of hoger voor we er iets van kunnen merken. Hieronder een afbeelding van het staart-effect na een 50-tal P-frames.

