

Verslag TMS

Image and Video Compressing

Kenneth Devloo 0623746
Nick Michiels 0623764

groep 2

1 Image Compressing

1.1 Algemeen

Voor image compression hebben we gekozen voor een GUI. Toch is het ook mogelijk om deze in console te gebruiken. Door in de console "--help" toe te voegen komt u te weten hoe de argumenten moet meegeven aan het programma. De GUI ziet er uit zoals in figuur 1.



Figuur 1 GUI van image compression

1.2 Instelbare parameters

Een *.raw afbeelding kan geopend worden via File->Open. Deze kan vervolgens op verschillende manieren worden gecompresseerd zo dadelijk worden toegelicht. Nadat de file gecompresseerd is kan er voor worden gekozen om ofwel deze in een gecompresseerd formaat op te slaan ofwel opnieuw als raw-file op te slaan. Dit kan door respectievelijk te kiezen voor File->Save Compressed Image of File->Save Raw Image.

Voor de compressie kunnen er 3 parameters worden ingesteld. Een eerste is blocksize die aangeeft hoe groot de kwantisatieblokken zijn. Hierbij kan een startwaarde gegeven worden voor de kwantisatie links bovenaan en een raise-waarde die de lineaire verhoging per diagonaal bepaalt.

1.3 Onze implementatie

Onze compressie gebeurt op de normale manier. Belangrijke punten waren voor ons:

- Rle wordt enkel op de nullen toegepast.

- Het zigzagalgoritme is snel en compact omdat deze veel gebruikt wordt.
- We gebruiken een matrixklasse die bewerkingen als dct,quantisatie en transponeren al ondersteunt om zo tot iets overzichtelijkere code te komen.
- We laten zoveel mogelijk parameters instelbaar.

2 Video Compressing

2.1 Algemeen

In tegenstelling tot image compressing hebben we bij video compressing enkel een console versie voorzien. Ook hier kan extra informatie betreffende de argumenten worden opgevraagd met "--help".

Voor video compressie hebben we vrij veel code kunnen hergebruiken uit de beeldcompressie. Dit is mogelijk gemaakt doordat we een specifieke klasse hebben die zich bezighoudt met de compressie van 1 afbeelding/frame. Voor de ganse I frame hebben we dus dezelfde aanroep kunnen doen die we gebruikt hebben voor 1 afbeelding in beeldcompressie (plus het toevoegen van headerinformatie). Verder hebben we ook zoveel mogelijk operaties met betrekking tot matrices gecentraliseerd in de matrixklasse. Bij alternatieve implementaties van compressiemethodes is het hergebruik van bijvoorbeeld een zigzag algoritme veel eenvoudiger. De P frames hebben we ook delen van de beeldcompressie kunnen hergebruiken. Wat betreft de referenties is er het zoekalgoritme bijgekomen, maar ook voor de verschilmatrices hebben we de beeldcompressiefuncties kunnen hergebruiken (hierbij is enkel een alternatieve functie van rle bij compressie en antiRle bij decompressie bijgekomen wegens het vastmaken van enkele parameters en dynamisch hergebruik van hetzelfde geheugen). De functie startCompression en startDecompression vormen de rode draad van het hele verhaal. Deze gaan dynamisch uit de file lezen en afwisselend naargelang de parameters makeIframe en makePFrame aanroepen.

2.2 Instelbare parameters.

Als u een video wilt comprimeren krijgt u typisch een uitdrukking in de vorm:

```
VideoCompress input.yuv output.y inColor height width GOP=1 maxMad=7 maxP=6  
macroblock=16 useDifference=0 blocksize=8 qstart=2 qraise=2
```

Zoals u ziet zijn er een aantal verplichte parameters en niet-verplichte parameters. Als eerste heb je een inputfile.yuv. Deze mag een kleuren video zijn of niet. Daarop volgend is er een output.y. Deze gaat de output-file zijn waar de gecompresse Y-band in zit. In de bool die daarop volgt (die je moet aanduiden met 0 of 1) kunt u kiezen of u in kleuren wil comprimeren of niet. Belangrijk om hierbij op te merken is dat wij expliciet gekozen hebben om de Y afzonderlijk te houden van de UV. We gaan dus effectief twee files creëren. Eén file output.y waar enkel de Y inzit en daarnaast een file output.uv die de gecompresse data van de U en V band bevat.

De drie volgende 5 parameters zijn specifieke instellingen voor videocompressie: Met de GOP kunt u de Group Of Pictures instellen, met de maxMad de maximale MAD-waarde voor het vinden van een referentie, MaxP stelt de maximale zoekruimte voor, met macroblock kunt u de grootte van de macroblock aanpassen en als u useDifference op 1 zet wordt er ook nog een verschilmatrix berekend voor de referentiemacroblokken.

De blocksize, qstart en qraise komen overeen met de parameters uit de beeldcompressie.

Om een video te decomprimeren hoeven er veel minder parameters mee gegeven te worden. De uitdrukking is als volgt:

```
VideoDecompress input.y output.yuv inColor removeArtefacts
```

Er moet dus enkel een inputfile en output file gegeven worden. Verder is het nodig te specificeren of u wil decompressen in kleur of niet. Zoals terjuist is toegelicht worden de kleuren gescheiden in een aparte *.uv file. Dit heeft als voordeel dat als er enkel de *.y file ter beschikking is onze decompressie er toch nog met overweg kan en in zwart-wit de video reconstrueert. Indien er naast de standaardfile ook nog de *.uv file bijstaat (met hetzelfde voorvoegsel als de *.y) kan de optie inColor worden aangezet en wordt deze informatie mee gebruikt. De laatste parameter is de removeArtefacts. Wij hebben in de loop van de implementatie opgemerkt dat er in de macroblokken linksboven witte pixels kunnen ontstaan. Dit is te danken aan afrondingen bij kwantisatie. Om dit op te lossen zijn we deze pixels gaan uitmiddelen met de omgeving. Met deze boolean kunt u er voor kiezen of u dit wil toepassen of niet.

2.2.1 Onze ervaringen

Door een beetje te spelen met de parameters kunnen we een vrij degelijke compressiemethode krijgen van 20%. Hiervoor moeten we de standaardwaarden nemen maar voor GOP 3, maxMad 5, maxP 5, blocksize 8, qstart 2 en qraise 3.

De maxP mogen we niet te hoog nemen omdat er anders verkeerde referenties kunnen gekozen worden. Maar met de huidige maxP wordt er toch al voor een groot procent van de macroblokken een referentie gevonden.

Ook de blocksize heeft een grote invloed op de mate van compressie. Blocksize van 8 geeft een heel degelijke compressie maar kan als nadeel geven dat de blokken met het blote oog zichtbaar kunnen worden. Indien de qstart op 1 zou worden gezet en qraise goed hoog worden gezet hebben we nog een vrij degelijke compressie van minder als 50 procent.

Omze decompressie is niet snel genoeg om af te spelen. Wij haalden 200 frames decompressie op 1 min. Compressie duurt iets langer.

2.3 Omze implementatie

- We laten weer zo veel mogelijk parameters instelbaar, dit is deels te danken aan de instelbaarheid van image compressie.
- Voor de motion vectoren gebruiken we het logaritmisch zoekalgoritme.
- Voor P-frames gebruiken we matrices van matrices waar mogelijk om overzicht te houden.
- Kleuren kunnen gecomprimeerd en opgeslaan worden in een aparte .uv-file, zonder p-frames.
- Het geheugen-verbruik is zo laag mogelijk gehouden (frame per frame verwerken, slechts enkele mb zo).
- Om rare artefacten te vermijden (witte ruis, pixel linksboven elk blok) door het afronden in de kwantisatie hebben we ervoor gezorgd dat deze pixels op de rand van de macroblokken uitgemiddeld worden met de pixels er rond.

2.4 Moeilijkheden

Wij hadden 2 punten waarop we lang vast zaten:

- Een bug in de imagecompressie waar het eerste getal in rle, indien dit een 0 was, als 0 keer 0 in de file kwam, wat dus al de rest kon in de war sturen.
- Een bug bij het maken van nieuwe buffers, waar we 1 byte te weinig mee gaven (afronding naar beneden in plaats van naar boven).