Implementatieplan

Imageshell

Jordy Alkema & Rick van Sambeek

4 April 2019

Inhoudsopgave

Doel

Het doel van dit implementatieplan is het besluiten van de beste manier voor het opslaan van een afbeelding, met als doel zorgen dat de conversie van RGB naar greyscale sneller word.

Methoden

Voor het opslaan van de afbeelding zijn er een aantal opties die wij moeten overwegen, de opties zijn:

Vector 1D

Een 1 dimensionale vector is een makkelijke manier voor het opslaan, alleen is dit iets moeilijker om te begrijpen omdat dit niet is hoe een afbeelding werkt in je gedachten. Het voordeel van het gebruik van een vector is dat je de maat dynamische kan aanpassen, dit levert alleen wel wat overhead op. 1D vectors zijn een veel voorkomend verschijnsel.

Vector 2D

Een 2 dimensionale vector is een makkelijke manier voor het opslaan, dit is makkelijker om in je hoofd te krijgen omdat een afbeelding ook 2D is. Ook deze is dynamische groter en kleiner te maken. 2D vectors zijn een veel voorkomend verschijnsel.

Linked list

Een linked list kan erg handig zijn omdat deze tot oneindigheid (Of je mogelijke ram) kan worden uitgebreid, het nadeel is dat dit wat moeilijker is om te implementeren en nog minder makkelijk te begrijpen is voor eventuele andere programmeurs.

Array 1D

Een 1 dimensionele array is een erg memory efficiënte manier van opslaan wanneer de afmetingen van een afbeelding vooraf bekend zijn. Omdat een array bij compileën al een grootte krijgt kan je dus niet achteraf de grootte bepalen. Verder is een voordeel dat iedere programmeur gebruik maakt van een 1D array waardoor dit makkelijk over te dragen is naar een andere programmeur.

Array 2D

Een 2 dimensionele array heeft dezelfde memory problemen als het 1D array. een voordeel is dat een 2D array een veelvoorkomend verschijnsel is in programmeren dus ook hier zouden eventuele andere programmeurs geen moeite mee moeten hebben. Een 2D array is nog minder efficiënt dan een 1D array, maar daarvoor terug krijg je wel wat meer duidelijkheid als programmeur omdat het makkelijker is om te debuggen.

Keuze

Criteria	Weging	1D Vector	2D Vector	Linked List	1D Array	2D Array
Snelheid	5	5	4	2	5	4
Efficiënt (RAM)	5	5	4	2	1	1
Implementatie	3	5	5	2	5	5
Leesbaar	3	3	5	2	3	5
Totaal		74	70	32	54	55

(0 = Slecht, 5 = Goed)

Onze keuze is gebaseerd op 4 factoren waarvan wij denken dat deze belangrijk zijn, Snelheid, Efficiënt, Implementatie en Leesbaar.

Uiteindelijk is het 1D vector geworden omdat deze het snelste is, goed te implementeren is. Een 1D vector is wat moeilijker te debuggen dan een 2D vector maar in verband met de eisen lijkt ons dat snelheid en efficiëntie belangrijker zijn, dit is hoe wij tot deze conclusie zijn gekomen.

Implementatie

Voor het implementeren van onze Imageshell gaan wij de volgende classen aanpassen:

- IntensityImageStudent
- RGBImageStudent

Wij gaan in deze classen een 1D vector implementeren, deze vector is vanaf de buitenwereld ook te benaderen als ‘2D Vector’.

Zodra er wordt gevraagd om een pixel op een X en Y positie, geven wij de pixel terug die op de locatie (Y * breedte van de afbeelding) + X in de vector staat.

Evaluatie

Wij gaan kijken of wij de goede conclusie hebben gemaakt door middel van het meten van de snelheid in verhouding met de standaard implementatie, en door middel van het RAM gebruik van de standaard implementatie.

Wij gaan deze testen doen omdat het doel is het converteren van RGB naar greyscale sneller te maken.

Bronnen

https://www.educba.com/c-plus-plus-vector-vs-array/

https://www.geeksforgeeks.org/linked-list-vs-array/

https://stackoverflow.com/questions/17259877/1d-or-2d-array-whats-faster