

1. Implementatieplan ImageShell

1.1. Namen en datum

Bas van Eck en Lennart Jensen. 26-3-2019

1.2. Doel

Het doel van deze implementatie is het ontwikkelen van een efficiënt en foutloos werkende datastructuur voor het opslaan van zowel RGB afbeeldingen als grayscale afbeeldingen. Deze datastructuur zal vervolgens geïmplementeerd worden in computer vision.

1.3. Methoden

Als eerste zullen we een implementatie maken die gebruikt maakt van `std::vector`. Vervolgens zullen we bepalen hoe lang het runnen dan duurt. Dit kunnen we vervolgens vergelijken met de tijd die de default implementatie nodig heeft om te runnen. Als blijkt dat onze implementatie veel langzamer is, kunnen we deze proberen te versnellen, bijvoorbeeld door direct op de heap te werken in plaats van `std::vector` te gebruiken als dit een snelheidswinst geeft.

1.4. Keuze

Bij de tests zijn beide datastructuren getest, als ook de default implementatie. Uit deze tests is gebleken dat de implementatie waarbij direct op de heap gewerkt wordt het meest geschikt is. Om die reden hebben we gekozen deze implementatie te gebruiken.

1.5. Implementatie

Door zowel de constructors als de set functies worden er voldoende bytes op de heap gealloceerd om de afbeelding van het gegeven formaat op te kunnen slaan. Verder worden door de destructor en de set functies eventuele bestaande data van de heap verwijderd. Als laatste worden zoveel mogelijk de eigen functies aangeroepen, om dubbele code te voorkomen.

1.6. Evaluatie

De ImageShells zijn getest op werking en snelheid, dit is vervolgens met elkaar vergeleken om tot een oordeel te komen over welke datastructuur het beste is.