# Effectiviteit volgorde filters

## Namen en datum

Alexander Streng groep 17 06-25-2014

## Doel

Tijdens het implementeren van de 3 filters (min, max & median) viel het me op dat de min & max filters op zichzelf niet voldoende hielpen om ruis op te ruimen. Aan de hand van dit meetrapport wil ik onderzoeken met welke combinatie en volgorde je de meest overeenkomende images krijgt.

## Hypothese

Ik vermoedt dat je het beste de min / max filters na de median kan toepassen om een plaatje te krijgen wat het meest overeen komt met de source (zonder ruis). Als de min / max voor de median worden gedaan zal de ruis leiden tot witte / zwarte kernels (3x3 blokken) in de image.

## Werkwijze

Om dit te testen zal ik op de volgende manier te werk gaan:

Ik kopieer de source image, en maak voor elke van de volgende een kopie. Aan die kopie voeg ik 5% ruis toe en pas ik de volgende filters in de gegevens volgorde toe:

* Min filter, Max filter, Median filter
* Min filter, Median filter, Max filter
* Max filter, Min filter, Median filter
* Max filter, Median filter, Min filter
* Median filter, Max filter, Min filter
* Median filter, Min filter, Max filter

Daarna zal ik voor alle images een vergelijking maken met de source image.

Ik zal de volgende image(512x512) gebruiken:

## Resultaten

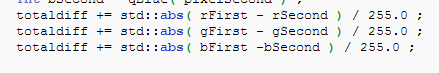
|  |  |
| --- | --- |
| Filter volgorde | Percentage % verschillend met origineel |
| Min filter, Max filter, Median filter | 4.45% |
| Min filter, Median filter, Max filter | 6.15% |
| Max filter, Min filter, Median filter | 4.30% |
| Max filter, Median filter, Min filter | 5.91% |
| Median filter, Max filter, Min filter | 3.10% |
| Median filter, Min filter, Max filter | 3.25% |

Zie volgende pagina voor de individuele resultaten.

## Verwerking

Het algoritme om het verschil in procenten te bepalen is vrij simpel;

Per pixel worden de waardes van de verschillende kanalen vergeleken en opgeslagen.



Om na de vergelijking het totaal te berekenen:



## Conclusie

Zoals redelijk verwacht is het het meest effectief om eerst een median filter over de ruis te halen alvorens de max / min algoritmen worden uitgevoerd. Dit is omdat het median algoritme de meeste ruis al zal verwijderen en de min / max dus niet de individuele ruispixels versterken. Opvallend genoeg is het ook efficiënter om eerst min , max en dan pas median uit te voeren in tegenstelling tot min, median max. Dit komt omdat de min / max filters elkaar ‘opheffen’

## Evaluatie

Toen ik mijn eerste implementatie van het min / max filter had uitgevoerd was ik nogal wat onzeker of de uitkomst ervan correct zou zijn. Het algoritme klopte, maar het leek mij een onwaarschijnlijk effect om uit te voeren op een beeld met ruis. Na het uitvoeren van deze experimenten is duidelijk geworden dat in combinatie met de tegenhanger van het filter (en misschien zelfs in combinatie met een median filter) we degelijke resultaten te behalen zijn. Jammer genoeg dekt deze test een heleboel onderdelen niet. Denk daarbij bijvoorbeeld aan het uitvoeren van de tests met enkele filters.

|  |  |
| --- | --- |
| Min filter, Max filter, Median filter | **F:\Projects\School\Vision-team17\Week 1 - 2\minMaxMedian-distance-to-original_4.45625%.png** |
| Min filter, Median filter, Max filter | F:\Projects\School\Vision-team17\Week 1 - 2\minMedianMax-distance-to-original_6.15155%.png |
| Max filter, Min filter, Median filter | F:\Projects\School\Vision-team17\Week 1 - 2\maxMinMedian-distance-to-original_4.30523%.png |
| Max filter, Median filter, Min filter | F:\Projects\School\Vision-team17\Week 1 - 2\maxMedianMin-distance-to-original_5.91974%.png |
| Median filter, Max filter, Min filter | F:\Projects\School\Vision-team17\Week 1 - 2\medianMaxMin-distance-to-original_3.10082%.png |
| Median filter, Min filter, Max filter | F:\Projects\School\Vision-team17\Week 1 - 2\medianMinMax-distance-to-original_3.25941%.png |