3e bachelor Informatica ICT
Universiteit HASSELT

2008-2009

Verslag Beeldverwerking

Two-scale Tone Management for Photographic Look

Nick Michiels 0623764
Jan Oris 0623977

1 Inleiding

Dit verslag gaat over de implementatie van Two-scale Tone Management for Photographic Look dat beschreven wordt in de paper (Bae, Paris, & Durand, 2006). Op deze website zijn ook handige alternatieve voorbeelden te vinden die we gebruikt hebben.

2 Gebruik van de CD-ROM

2.1 Mapindeling

Er zijn 3 mappen aanwezig op de cd-rom: de map 'doc' bevat dit verslag, de map 'src' bevat al de '.m'-files en de map 'examples' geeft een aantal alternatieve afbeeldingen met hun output.

2.2 Project runnen

Ons basisproject is opgebouwd voor RGB-kleurenafbeeldingen. De executabel-file om de standaardinput met standaardmodel uit te voeren is 'Project'. Bovenaan deze file kunnen eventueel andere input en model worden gespecificeerd. Per stap in het project worden er tussenliggende afbeeldingen opgeslagen. Deze komen terecht in een map 'resultImages' in de work-map van het project. Het is mogelijk dat het project een error geeft, dit kan liggen aan het feit dat de map niet bestaat. In de 'src'-map staat er een voorbeeldmap van hoe de 'resultImages'-map er zou moeten uitzien.

3 Structuur

3.1 Algemeen

Voor onze structuur hebben we gekozen om kleuren en monotone afbeeldingen strikt gescheiden te houden. U zal zien dat zo goed als elke functie uit typisch twee exemplaren bestaat: een gewone 'functie' en een 'functieMonoColor'. De mono color functies dienen typisch voor de toepassing op 1 kleurenband. Dit kan R, G of B zijn maar ook bijvoorbeeld L in de CIE-LAB kleurenvoorstelling. De gewone functie werkt met de RGB kleuren en zal gegeven een 3D-RGB-kleurenafbeelding deze afbeelding opsplitsen in de 3 kleurenbanden en deze meegeven aan de respectievelijke 'MonoColor'-functie.

Verder kunt u in de executabel-file de verschillende stappen uit de opgave/paper duidelijk onderscheiden. Voor elke stap wordt er zijn eigen functie aangeroepen. Deze executabel gebruikt standaard de functies voor RGB-afbeeldingen. Wit u nu zelf een grijswaarden afbeelding of een andere kleurenband als input geven, kunt u de functie 'ProjectMonoColor' gebruiken. Let op, dit is wel een gewone functie waarmee parameters moeten worden meegegeven (hierbij moeten de kleurenwaardens binnen het bereik [0,1] liggen).

Dit gezegd zijnde kan er nu wat dieper ingegaan worden op de interne structuur. In de volgende secties gaan we elke grote stap in de implementatie wat meer toelichten.

3.2 (Cross) Bilateral Filter

Onze bilaterale filter is een fast bilateral filter van (Paris & Durand, 2007). Onze implementatie is gebaseerd op de formules van pagina 11 uit de paper (zie Figuur 1). De automatische toekenning van sigmaS en sigmaR hebben we ook opgenomen uit deze website.

Over het algemeen bestaat deze snelle bilaterale filter uit 3 grote stappen.

1. Ten eerst wordt de afbeelding gedownsampled. Door het downsamplen zal de afbeelding veel kleiner worden waardoor er veel minder operaties zullen moeten worden uitgevoerd. Voor dit samplen hebben we een sampleR en sampleS nodig. Deze kunnen we gelijk stellen als de SigmaR en sigmaS.

2. De tweede stap bestaat er uit om op de gedownsampelde matrix een convolutie te doen van een gausiaanse filter met sigmaS.

3. Nadat de filter is toegepast rest er ons enkel nog terug te upsampelen naar de originele grootte.

De performantie in deze aanpak zit em erin dat de convolutie nu op een veel kleinere schaal gebeurd waardoor er veel minder berekeningen nodig zijn.

FAST BILATERAL FILTER

input: image I
Gaussian parameters σ_s and σ_r
sampling rates s_s and s_r

output: filtered image I^b

1. Initialize all $w_\downarrow$ $i_\downarrow$ and $w_\downarrow$ values to 0.
2. Compute the minimum intensity value:

$$I_{\min} \leftarrow \min_{(X,Y)\in\mathcal{S}} I(X,Y)$$

3. For each pixel $(X,Y) \in \mathcal{S}$ with an intensity $I(X,Y) \in \mathcal{R}$
   (a) Compute the homogeneous vector (wi, w):

$$(wi, w) \leftarrow (I(X,Y), 1)$$

   (b) Compute the downsampled coordinates (with $\lfloor \cdot \rfloor$ the rounding operator)

$$(x, y, \zeta) \leftarrow \left(\left\lfloor \frac{X}{s_s} \right\rfloor, \left\lfloor \frac{Y}{s_s} \right\rfloor, \left\lfloor \frac{I(X,Y) - I_{\min}}{s_r} \right\rfloor\right)$$

   (c) Update the downsampled $\mathcal{S}\times\mathcal{R}$ space

$$\begin{pmatrix} w_\downarrow\, i_\downarrow(x,y,\zeta) \\ w_\downarrow(x,y,\zeta) \end{pmatrix} \leftarrow \begin{pmatrix} w_\downarrow\, i_\downarrow(x,y,\zeta) \\ w_\downarrow(x,y,\zeta) \end{pmatrix} + \begin{pmatrix} wi \\ w \end{pmatrix}$$

4. Convolve $(w_\downarrow\, i_\downarrow, w_\downarrow)$ with a 3D Gaussian g whose parameters are σ_s/s_s and σ_r/s_r

$$(w^b_\downarrow\, i^b_\downarrow, w^b_\downarrow) \leftarrow (w_\downarrow\, i_\downarrow, w_\downarrow) \otimes g$$

5. For each pixel $(X,Y) \in \mathcal{S}$ with an intensity $I(X,Y) \in \mathcal{R}$
   (a) Tri-linearly interpolate the functions $w^b_\downarrow\, i^b_\downarrow$ and $w^b_\downarrow$ to obtain $W^b\, I^b$ and W^b:

$$W^b\, I^b(X,Y) \leftarrow \text{interpolate}\left(w^b_\downarrow\, i^b_\downarrow, \frac{X}{s_s}, \frac{Y}{s_s}, \frac{I(X,Y)}{s_r}\right)$$

$$W^b(X,Y) \leftarrow \text{interpolate}\left(w^b_\downarrow, \frac{X}{s_s}, \frac{Y}{s_s}, \frac{I(X,Y)}{s_r}\right)$$

   (b) Normalize the result

$$I^b(X,Y) \leftarrow \frac{W^b\, I^b(X,Y)}{W^b(X,Y)}$$

Figuur 1 Formules die we gebruikt hebben voor onze implementatie van de fast bilateral filter. De formules zijn afkomstig uit (Paris & Durand, 2007)

3.3 Gradient Reversal Removal

U zal het misschien vreemd vinden dat er voor dit zoveel verschillende functies aanwezig zijn. Hiervoor zijn twee verklaringen. Ten eerste zoals we zonet al hebben aangehaald is er overal een opsplitsing in mono color en rgb. Maar naast deze opsplitsing hebben we ook nog een extra opsplitsing gedaan. Namelijk er wordt niet enkel op de baseImg een oproep gedaan op GradientReversalRemoval, maar in de laatste stap die nog moet komen, de Detail Preservation, wordt er ook een aanroep gedaan naar gradient reversal. Het verschil hierbij is dat de implementatie anders moet. Samengevat komen we dus op 4 functies terecht.

3.4 Histogram Matching

Voor de implementatie van histogram matching hebben we het voorbeeld gevolgd uit de boek (Gonzalez & Woods, 2008).

Een belangrijke opmerking hierbij is dat wij er voor gekozen hebben eerst een equalisation uit te voeren alvorens de effectieve specification. Dit zou volgens het boek een beter resultaat geven als direct een specification.

3.5 Textureness

Textureness is onze traagste stap in het proces. Hij bestaat uit drie delen. Als eerste berekenen we met onze functie 'HighPassFilter' een gefilterde afbeelding van zowel inputImage, modelImage, histogramBaseImage en DetailImage. Vervolgens herbruiken we onze cross bilaterale filter waarbij we de gefilterde afbeeldingen als input meegeven en de originele als edge. De laatste stap wordt uitgevoerd in een aparte functie, namelijk de 'TexturenessTransfer'-functie. Hierin word de overeenkomstige berekening van de paper uitgevoerd.

3.6 Detail Preservation

De laatste stap van het project is het trachten om terug meer detail te krijgen in het beeld. Ook deze implementatie heeft zijn eigen functie met de overeenkomstige naam 'DetailPreservation'.

4 Extra

4.1 CIE-LAB

Wij hebben het derde additional effect van de paper toegevoegd aan onze implementatie. Namelijk de Color and Toning in de CIE-LAB space. Op deze manier hoeven we enkel ons project toe te passen op de L-layer. De a en b kunnen we behouden.

Deze extra heeft ook zijn eigen executabel, genaamd 'projectLAB'. Net zoals bij de vorige kan bovenaan deze file de standaard parameters worden aangepast zoals input en model. Het resultaat van deze uitvoering wordt vergeleken in Figuur 2.

Zoals je ziet maakt projectLAB gebruik van de algemenere functie LAB die alternatief kan gebruikt worden.

Figuur 2 De bovenste afbeelding is de input, de middelste het resultaat van de site van de paper, de onderste is ons resultaat

5 Algemene Opmerkingen

- Indien u functies gebruikt moet u altijd opletten dat u afbeeldingen meegeeft met kleurwaarden binnen het interval [0,1].

- In sommige versies van MatLab werkt de interne ‘prctile’-functie niet. Hiervoor hebben we ook een eigen implementatie, namelijk ‘Percentile’. We gebruiken deze eigen implementatie niet omdat in sommige gevallen deze echter niet performant blijkt te werken (en er zelfs niet

uitgeraken).
Indien u toch deze eigen implementatie wil gebruiken (die zeker werkt op de standaard input en standaard model), hoeft u enkel regel 27 van de 'DetailPreservationMonoColor.m' en/of regel 27-28-29 van 'DetailPreservation.m' door de 'prctile' te vervangen door 'Percentile'.

- In elke mono color van de 'GradientReversalRemoval'-functies bevindt zich er een speciale boolean 'jan' die standaard op false staat. Deze staat er wegens het feit we een eigen implementatie van gradientberekening hebben en een matlab implementatie. Standaard wordt onze eigen implementatie gebruikt (jan=false), maar als deze op true staat wordt de matlabfunctie 'gradient()' gebruikt die we later te weten gekomen zijn via onze medestudent Robin Marx.

6 Referenties

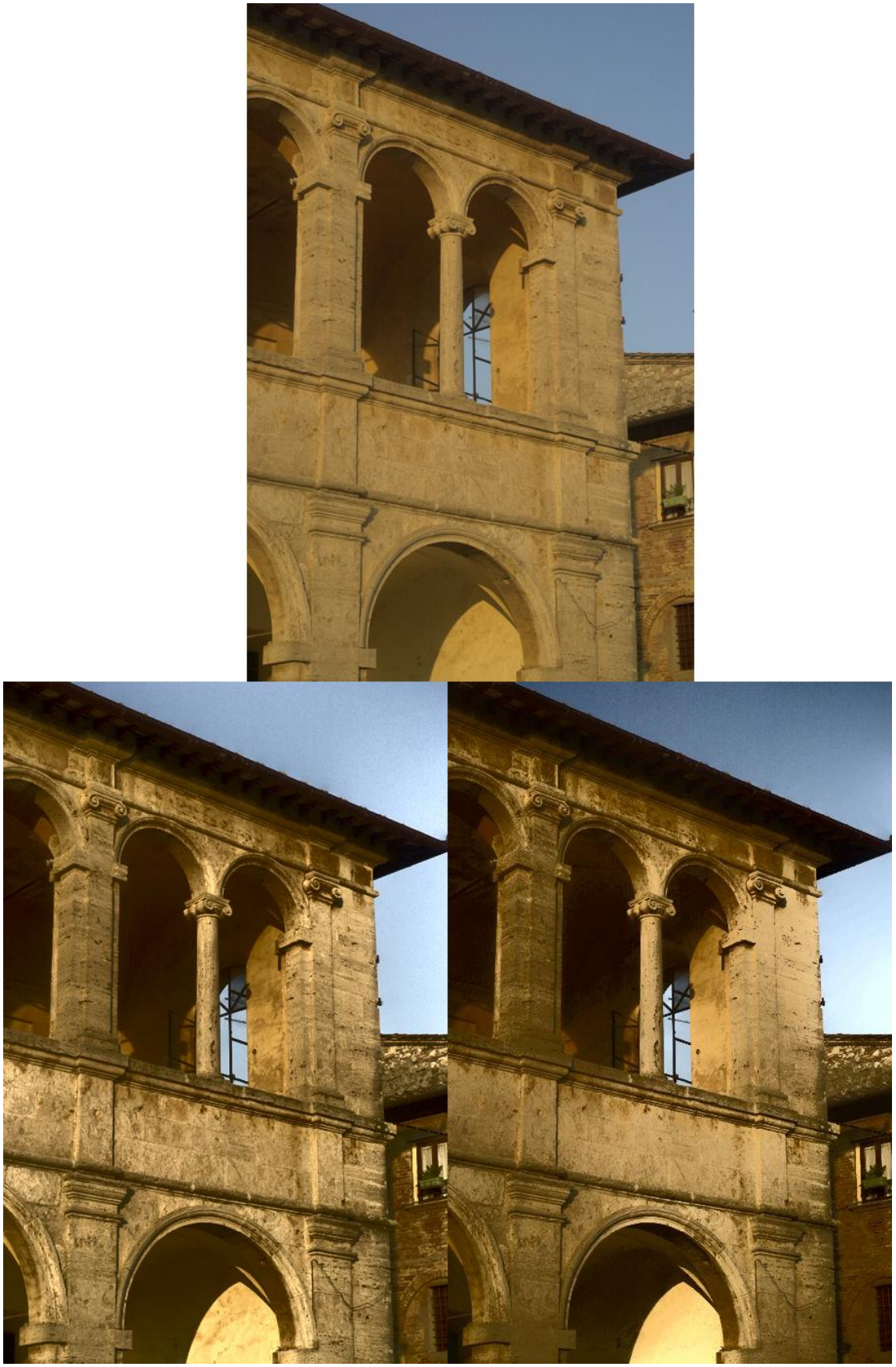
Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing. In R. C. Gonzalez, & R. E. Woods, *Digital Image Processing* (pp. 120-139). New Jersey: Pearson Education, Inc.

Paris, S., & Durand, F. (2007, Maart 2). *Fast Bilateral Filter*. Opgehaald van http://people.csail.mit.edu/sparis/bf/

7 Voorbeelden

Omdat de meeste voorbeelden van op de website van de paper gemaakt zijn in de CIE-LAB color space zijn onze voorbeelden ook gemaakt in dezelfde color space om ze beter te kunnen vergelijken.

Figuur 3 Ons standaardproject na het runnen van de 'Project' executabel. Bovenaan is de input, links onder is het resultaat dat op de didactiek te vinden is, rechts onder is ons resultaat.



Figuur 4 Bovenste afbeelding is input, afbeelding links beneden is voorbeeld van de website van de paper, afbeelding rechtsonder is ons resultaat

Figuur 5 De bovenste afbeelding is de input, de middelste het resultaat van de site van de paper, de onderste is ons resultaat