IIIF : Deliverable 4 - Image processing workflow

Bart Debunne
bart.debunne@meemoo.be
version 3, 01-10-2021

Inhoudsopgave

Doelstelling

Specificaties

Afgeleide beeldbestanden

Caution

Voor afgeleide bestanden moet onderscheid worden gemaakt tussen werken in public domain en werken die nog onder het auteursrecht vallen. In de overeenkomst die de VKC momenteel met de auteursrechtenorganisatie Sabam heeft, worden volgende hergebruiksvoorwaarden vooropgesteld voor werken die onder het auteursrecht vallen: de resolutie van het gereproduceerde beeld mag niet meer dan 640x480 pixels zijn en een resolutie van maximum 72dpi hebben. Onderstaande specificaties hebben betrekking op werken die niet onder die overeenkomst vallen.

- jpeg-2000 (jp2)
- 300 ppi
- sRGB
- Bestanden worden bijgesneden (*crop*), randinformatie zoals kleurkaarten en kaders verwijderd

- Embedded metadata(XMP, Exif, ICC) wordt indien aanwezig overgenomen van het origineel

We testen de workflow aan de hand van een representatieve referentieset van ca. 200 beelden. Deze set bevat de beelden uit fase 1 van het VKC IIIF-project aangevuld met een aantal werken van James Ensor.

Relatief herschalen

De te ontsluiten collectie is zowel qua type als qua fysieke afmeting zeer divers, voor de hele VKC-collectie spreken we over heel kleine objecten van een paar centimeter tot wandtapijten, of een wandkaart van ettelijke meters hoog en breed. De afmeting in pixels van het digitale beeld is niet per se in verhouding en is afhankelijk van het moment van de opname, de fotograaf, het opnametoestel, etc. Voor sommige grote werken zijn verschillende foto's aan elkaar geplakt (stitching). Andere zijn gevat in 1 foto. Relatief gesproken kan een groter fysiek werk in aantal pixels kleiner of gelijk zijn aan een fysiek kleiner werk.

Grote bestanden betekenen echter een grotere relatieve kost aan opslag, bandbreedte en verwerking. Daarom dringt zich, zeker bij een groot aantal beelden, een beperking van de bestandsgrootte op, i.c. herschalen van de beelden naar een aanvaardbare grootte met minimaal kwaliteitsverlies.

Bij vaste herschaling wordt de langste of kortste zijde herleid tot een vooraf bepaalde waarde. Waarbij het resulterend bestand voldoende detail bevat en tegelijk de performantie waarborgt. Bij aanvang van het project werd dit bepaald als (max.) 5000 pixels voor de kortste zijde. Voor bestanden die merkelijk groter zijn dan die bovengrens betekent dit echter een veel ingrijpender herschaling met potentieel matig tot goed merkbaar kwaliteitsverlies. Dynamisch herschalen probeert deze impact te mitigeren door progressief te herschalen, eventueel in verhouding tot de fysieke grootte van het werk.

Om te vermijden dat (heel) grote werken te fel herschaald zouden worden ten opzichte van kleine werken met een in verhouding hoger aantal pixels, is daarom binnen dit project bekeken in welke mate een relatie bestaat tussen de fysieke en digitale afmetingen en of op basis daarvan de herschaling relatief of progressief kan gebeuren.

Aan de hand van de referentieset is gekeken of er een relevante correlatie is tussen de grootte van een afbeelding en de fysieke afmeting van een werk en of het al dan niet nuttig is rekening mee te houden bij het herschalen van het digitale beeld.

Voor 77 testbestanden uit de referentieset werden de fysieke met de digitale afmetingen statistisch vergeleken in R. Alleen de breedte werd in aanmerking genomen om te compenseren voor afwijkingen door mogelijke kleurenkaartstroken onderaan de afbeelding.

```
Sample: 77
```

```
Mean width in cm: 90
Mean width in px: 6169
cor (pearson): 0.09

data:  cm and px
t = 0.80326, df = 75, p-value = 0.4244
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1344031  0.3099238
sample estimates:
       cor
0.09235595
```

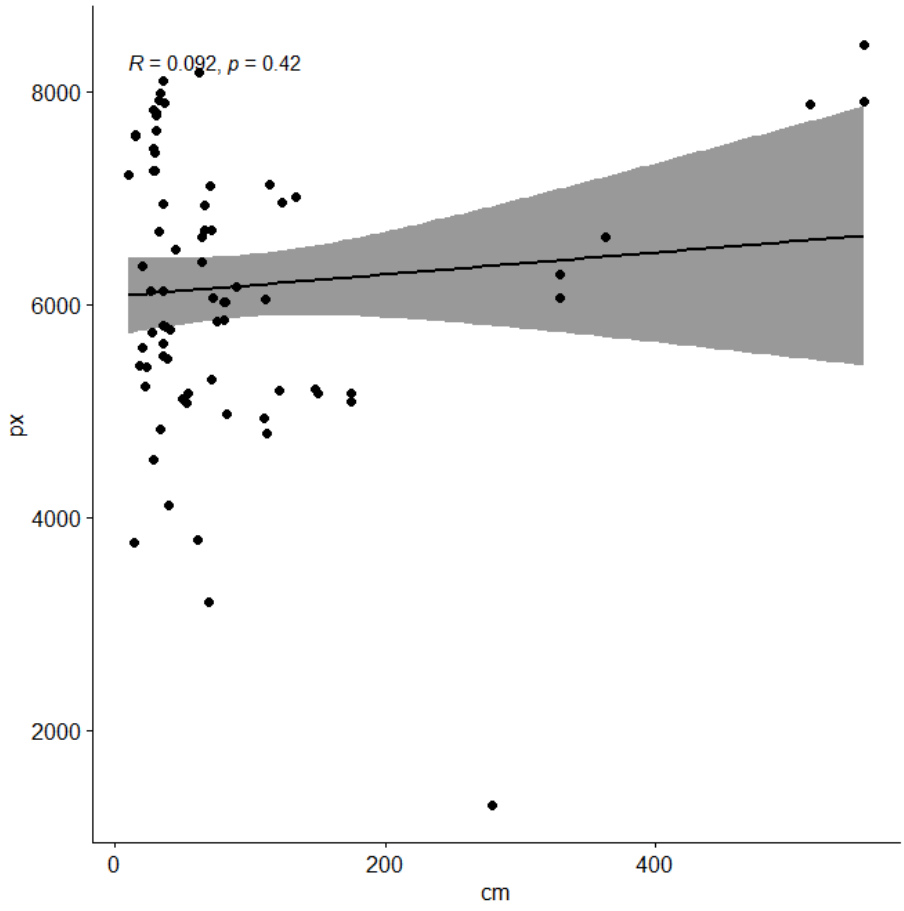

Figure 1: Graph

Resultaat:

- gemiddelde breedte = 90 cm
- gemiddeld aantal pixels = 6169px
- geen correlatie tussen fysieke breedte en breedte in pixels: 0.09

Dit is slechts een kleine set met een slechte p-value! Een zelfde vergelijking werd daarom uitgevoerd op de volledige set aan beelden van VKC die beschikbaar zijn op Art in Flanders (AIF), met een gelijkaardig resultaat.

```
Sample: 12639
Mean width in cm: 98
Mean width in px: 6024
cor (pearson): 0.23

data:  cm and px
t = 26.744, df = 12639, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2148652 0.2478637
sample estimates:
     cor
0.231431
```

Resultaat:

- gemiddelde breedte = 98 cm
- gemiddeld aantal pixels = 6024px
- weinig tot geen correlatie tussen fysieke breedte en breedte in pixels: 0.23

Conclusie:

De resolutie clustert rond een mediaan van 6050 pixels, met het gros tussen 4000 en 8000 pixels, afnemend in aantal boven 10.000px tot enkele zeldzame pieken boven 20.000px (max. 25k). Er is geen waarneembare correlatie tussen de fysieke afmeting van beelden en de resolutie in pixels.

Er wordt voorgesteld de fysieke afmeting los te laten en de digitale afmeting als leidend te zien. Om alsnog een balans tussen opslagcapaciteit, performantie en kwaliteit te waarborgen wordt volgende getrapte herschaling voorgesteld:

- Afbeeldingen tot 5.000px breedte worden ongemoeid gelaten
- Van 5.001-10.000px 50% herschalen
- Max. breedte 10.000px

(images)

De referentieset is opnieuw gehanteerd als representatieve controleset om de impact van het herschalen na te gaan.

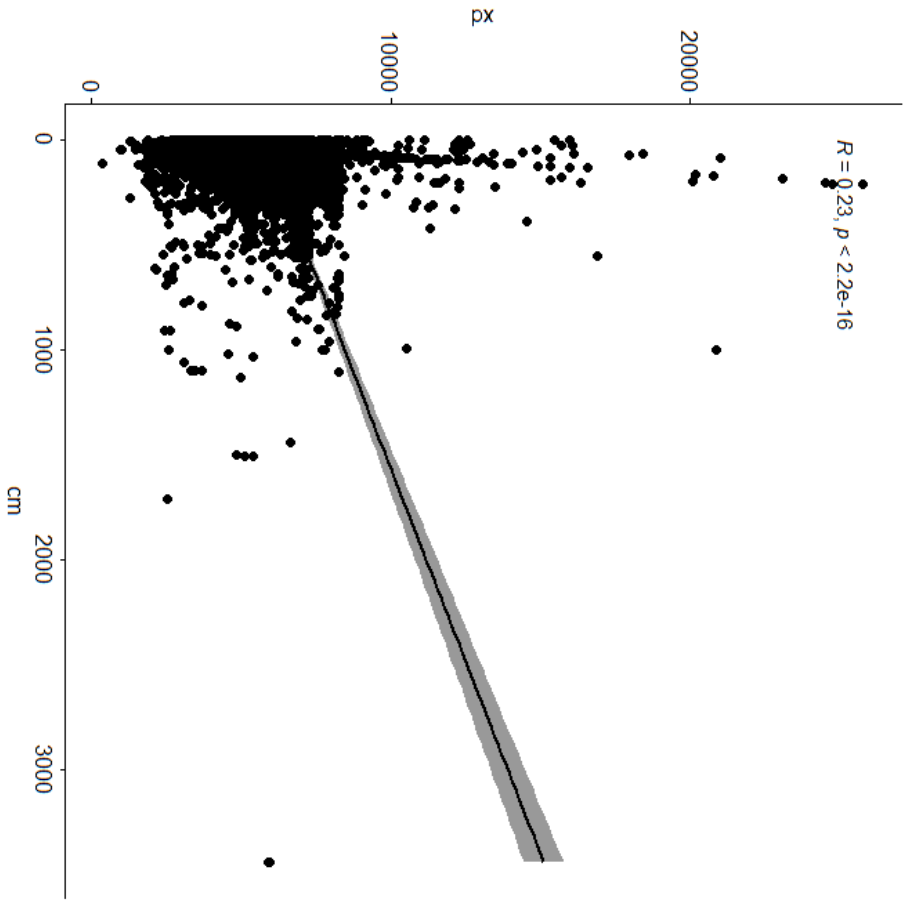

Figure 2: Graph

Formaat

Totale bestandsgrootte

Archiefmaster TIFF

37,92 GB

Afgeleide JPEG2000 - niet herschaald

5,95 GB

Afgeleide JPEG2000 - “dynamisch” herschaald

4,92 GB

De totale omvang van de afgeleide beelden is respectievelijk 15,5% en 13% van die van de originele archiefbestanden. Het verschil tussen herschalen of niet herschalen levert een besparing van ongeveer 20% op aan opslagcapaciteit: ca. 1GB minder per 200 beelden.

Er is visueel een kwaliteitsverlies waarneembaar bij het inzoomen. Bij de grootste beelden die verhoudingsgewijs meer herschaald zijn, is een verzachting van de contouren merkbaar. Het kwaliteitsverlies is progressief in die zin dat een origineel beeld van 15000px breed relatief meer effect zal ondervinden van herschaling dan een beeld van 7000px breed.

Hoewel er geen detail of kleur verloren lijkt te gaan, zijn de edges en details minder scherp waardoor de afbeelding minder crisp over komt op de hogere zoomniveau’s. In de hoofdzoom is geen verschil waar te nemen.

Eindconclusie

Het “dynamisch” herschalen levert een gemiddelde besparing in opslag van 20%. Voor grotere bestanden komt dit met een matig en progressief kwaliteitsverlies, waarneembaar bij de diepere zoomniveau’s als een “verzachting” van de details in het beeld.

Voor materiaal gelijkaardig qua resolutie aan de referentieset is de resulterende kwaliteit aanvaardbaar. Indien nog grotere beelden worden aangeboden dienen de drempelwaardes voor het herschalen echter herbekeken te worden.

Het lineair herschalen tot een grens van 5000px, zou een grotere degradatie van de kwaliteit betekenen voor beelden boven en is niet wenselijk.

Taken

De workflow kan gezien worden als een ETL-proces. Hierin worden min of meer procedureel de volgende taken uitgevoerd:

- Haal bestand op van tape aan de hand van een id
- Detecteer randinformatie en snij bij indien nodig

- Herschaal tot bepaalde grootte
- Zet om naar sRGB kleurprofiel
- Sla op als gecomprimeerde jp2 inclusief originele metadata
- Analyseer en valideer resultaat
- Verplaats naar eindbestemming
- Kuis tussenbestandenop

Export (*E*TL:extract)

In deze stap worden de originele archiefmasters uit het archiefsysteem (MAM) geësxporteerd via de export API van het MAM. De ongecomprimeerde hogeresolutie tiff-bestanden worden hierbij van tape naar lokale disk in de VM gekopieerd. Een export kan geïnitieerd worden voor 1 of meerdere bestanden ineens.

Voor het exporteren van een esssence is een Mediahaven fragment-id nodig. Indien niet bekend kan die eerst worden opgehaald aan de hand van de meemooidentifer (pid) of overige lokale id met een metadata query naar de REST API.

Voor meer informatie over de REST API van MediaHaven en het exporteren van essences zie de Mediahaven REST API manual.

Transformatie (E*T*L:transform)

Detecteer en verwijder kleurenkaart Voor het detecteren van de kleurenkaart maken we gebruik van de colorchecker zoals ontwikkeld voor het Flore de Gand project. De colorchecker gaat op zoek naar een kleurenkaart en als die wordt herkend dan wordt de strook met de kleurenkaart weggesneden en de afbeelding opgeslaan.

De colorchecker zou ook in staat moeten zijn om kaders en randen te detecteren, zoals een passe-partout of lijst. Indien gewenst zou het in een volgende fase mogelijk zijn om een model te trainen om ook dit bij te snijden.

> The code is such that, when an input image is provided, it would perform detections and crop out the colour scale so that only the painting would be saved.

Code: https://github.com/tckrishna/colorchecker.git Documentatie en tutorial op: https://colab.research.google.com/drive/1OreAxCrCTkTqbIxu2Z_8KWuCujKyyncI?usp=sharing

Omdat de oorspronkelijke code moeilijk overweg kon met 16-bit bestanden en de kleurenkaart verbergt onder een wit vlak ipv deze volledige weg te snijden heeft meemoo een paar aanpassingen gedaan in een eigen fork.

Herschaal De afbeelding worden dynamisch herschaald cfr. supra Relatief herschalen.

De afmeting wordt herschaald op basis van de kortste zijde:

- < 5000px: niet herschalen
- > 5000px: herschalen naar (5000 + zijde - 5000) /2
- > 15000px: herschalen naar 10000

Voorbeelden:

- 3650 => 3650
- 6200 => 5600
- 10000 => 7500
- 17000 => 10000

Code: https://github.com/viaacode/iiif-image-processing/blob/main/app/helpers.py#L128

Detecteer en stel kleurprofiel (ICC) in We detecteren het kleurenprofiel van het om te zetten beeldbestand en indien afwezig of anders wordt het omgezet naar `sRGB`.

Sla op als jpeg2000 Voor het comprimeren en opslaan als jp2 gebruiken we de gelicensieerde Kakadu software. We gebruiken het profiel van Digital Bodleian: https://github.com/viaacode/iiif-image-processing/blob/main/app/file_transformation.py#L97

```
kdu_compress -i input.tif -o output.jp2 Clevels=6 Clayers=6 "Cprecincts={256,256},{256,256},
```

Important

The JPEG 2000 format supports only a restricted set of ICC Profile features. The -jp2_space parameter on kdu_compress sets the colour profile in the image metadata, but does not otherwise convert the image - the pixel values remain the same. The sRGB value sets the colour profile to the sRGB IEC61966-2.1 profile. (This is not the only way to set the colour profile) Kakadu (and JP2 itself) will not support CYMK images: Only three colour channels, R (red), G (green) and B (blue), are supported by the JP2 file format. For example the sRGB v4 ICC preference profile is not supported, and cannot be embedded into a JP2 file using Kakadu. Setting -jp2_space sRGB on kdu_compress will erase the embedded profile and so allow it to be converted. The sRGB IEC61966-2.1 profile thus assigned is sufficiently different that in some cases there is a noticeable tint to the created JP2. https://readthedocs.org/projects/image-processing/downloads/pdf/latest/

Het instellen van de sRGB color space in de vorige stap voorkomt problemen (afwijkende kleur) gerelateerd aan de expliciete toekenning van de sRGB space in kdu_compress.

Kopieer metadata kdu_compress kopieert niet alle metadata tags.

Met behulp van een tool als. *exiftool* worden alle embedded metadatatags zoals XMP en IPTC uit het origineel gelezen en gekopieerd naar het afgeleide bestand. Onderstaand commando bijvoorbeeld geeft alle XMP en IPTC tags gegroepeerd per *tag family* terug in JSON-formaat.

```
$ exiftool 7659c97c0w.tif -XMP:All -IPTC:All -g0:1 -json
```

Op deze manier kan ook metadata cleaning worden gedaan van bron naar afgeleide. Het kan bijvoorbeeld nuttig zijn redundante of verouderde en niet (langer) relevante tags eerst te verwijderen. Het is eveneens mogelijk in deze stap tags toe te voegen op basis van metadata uit het MAM of de VKC-databronnen.

Zie voor meer informatie de Exiftool website en de Exiftool manual met voorbeelden.

Meer info over IPTC: http://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata

Valideer Valideer dat het eindresultaat voldoet aan de volgende assertions:

- ppi = 300
- icc = sRGB
- metadata tags = source file tags
- file format = valid jp2
- file name = pid

https://jpylyzer.openpreservation.org https://github.com/openpreserve/jpylyzer https://exiftool.org/index.html https://exiftool.org/exiftool_pod.html

Verplaats (ET*L*:load) Bestanden worden naar de eindbestemming gekopieerd waar ze steekproefsgewijs visueel geïnspecteerd kunnen worden. Als bestandsnaam wordt de meemoo pid (external_id) gebruikt en `.jp2` als extensie.

De eindbestemming is een folder die de media mount point is voor de IIPImage server.

Ruim op Tussentijdse bestanden en met succes verwerkte bronbestanden worden verwijderd. Gefaalde bestanden blijven staan voor inspectie.

Workflow

Figure 1. Voorbeeld manuele workflow voor creatie van jp2 afgeleide beeldbestanden

Voor de creatie van de afgeleiden starten we met een vrij manuele workflow die eenvoudig kan bijgesteld worden om uiteindelijk te komen tot een automatiseerbare workflow. Om zowel de workflow voor de creatie van afgeleide beelden als de specificaties an sich te testen beperken we ons in eerste instantie tot de omzetting van de beelden die nu reeds beschikbaar zijn in de IIIF-viewer in de VKC Arthub. Hierbij zal worden onderzocht welke een haalbare workflow is voor de aanmaak van de afgeleide beeldbestanden en in welke mate dit proces geautomatiseerd kan worden. Indien nodig kunnen bovenstaande specificaties dan ook bijgewerkt worden op basis van voortschrijdend inzicht.

Tech specs

Omgevingen

DEV: lokale omgeving bij dev QAS en PRD: Debian VM + data store (disk)

Deployment via Puppet/Foreman en parametriseerbaar.

Stack

- Taal: Python
- Metadata read/write: exiftool
- jp2 schrijven: kdu_compress (kakadu)
- jp2 validatie: jpylyzer

Execute

Manueel.

In latere fase automated trigger op basis van metadata attribuut in MAM.

Observability

Single line json logging naar stdout => ELK.