

---

# Meetrapport snelheid conversie

---

Florian Humblot - 1720570  
Vera Schoonderwoerd - 1721202

Datum: 22 februari 2019

## Inhoudsopgave

Doel .....	3
Hypothese .....	3
Werkwijze.....	3
Resultaten .....	4
Verwerking .....	5
Conclusie .....	6
Evaluatie .....	6

## Doel

Het doel van dit meetrapport is het testen hoe veel sneller onze implementatie is, ten opzichte van de default implementatie. Dit onderzoek is met betrekking van zowel onze container als onze intensity conversie.

## Hypothese

Wij verwachten dat onze implementatie 30% sneller zal zijn, omdat onze implementatie maar met 1 RGB kleur rekent. Wij verwachten dat de default conversie namelijk met 3 RGB waarden rekent.

## Werkwijze

Deze test gaan wij met 7 verschillende fotos doen. Per foto wordt het inladen van de foto, met onze zelfgemaakte methode, en het converteren naar intensity 100 keer aangeroepen: 50x met de default methode en 50x met onze eigen methode. De code is in debug-mode gerunned, zonder flag optimalizaties. Om de tijd bij te houden gebruiken wij de functie `std::clock()`. De beginwaarden en de eindwaarden halen we van elkaar af en dit delen we door (`CLOCKS_PER_SEC/1000`). Hieruit volgt de conversietijd in milliseconden. Deze waarden schrijven we naar een CSV bestand, om de resultaten makkelijk naar Excel te kunnen converteren. Als alle 700 conversies gedaan zijn, gaan we het gemiddelde berekenen per foto per methode.

## Resultaten

De resultaten van het experiment zijn in onderstaande tabel weergegeven. Zoals te zien is, zijn er per foto 50 conversies geweest van de default methode en 50 conversies van onze eigen implementatie. De default implementatie is rood gekleurd, de studentversie is groen gekleurd.

child-1 default	child-1 student	female-1 default	female-1 student	female-2 default	female-2 student	female-3 default	female-3 student	male-1 default	male-1 student	male-2 default	male-2 student	male-3 default	male-3 student
939	403	420	367	174	131	362	305	338	303	606	368	392	350
729	436	410	359	155	147	348	307	331	300	650	430	386	344
631	404	395	386	154	160	346	305	334	295	477	380	391	339
508	408	403	361	149	143	347	314	345	303	523	425	373	345
567	411	408	365	156	139	356	304	350	291	503	347	389	336
475	399	403	365	158	157	341	303	336	331	431	350	390	360
447	407	401	368	156	144	347	314	339	295	416	374	391	347
419	349	415	364	162	130	352	301	330	297	419	349	378	340
416	364	408	375	157	143	345	319	352	291	400	336	401	336
390	352	408	356	150	151	349	296	339	288	396	339	401	360
401	366	405	359	161	136	351	316	346	318	402	341	380	526
397	346	403	355	160	143	344	303	326	298	408	340	394	471
396	366	402	370	154	149	351	308	344	295	420	336	382	472
397	380	422	352	161	147	353	317	337	294	397	323	402	464
389	347	406	347	158	140	348	302	330	298	400	370	398	470
397	340	418	361	159	142	339	305	335	286	401	367	383	361
399	367	409	374	153	143	356	317	348	292	397	334	385	360
395	370	402	359	151	132	351	311	334	289	404	343	386	341
388	351	402	359	153	137	347	316	337	285	406	354	374	338
419	359	401	354	156	144	348	303	328	311	412	349	388	327
453	372	411	308	155	141	362	316	334	302	411	346	396	293
443	362	407	294	155	146	346	311	335	297	404	359	394	283
438	349	407	306	150	145	344	314	345	287	407	339	386	288
416	356	411	302	163	149	348	306	337	296	399	357	383	295
414	367	411	307	149	152	377	299	350	294	403	342	398	303
447	334	411	300	162	138	344	301	337	306	414	330	378	295
418	364	410	302	155	154	350	313	341	299	413	345	378	285
396	349	393	305	153	137	345	297	347	309	406	380	394	287
402	378	396	310	157	140	351	319	336	307	399	338	386	284
418	348	410	315	155	141	349	303	347	298	503	327	367	313
408	361	403	316	154	136	362	300	339	292	512	331	385	287
401	368	404	297	159	150	347	331	331	291	444	356	410	289
404	395	404	301	169	148	360	299	363	297	473	341	375	283
447	357	412	303	151	166	346	296	347	302	453	346	392	285
454	363	410	315	149	138	335	305	337	309	425	340	384	303
476	358	415	300	153	158	350	312	339	288	411	350	382	280
473	349	416	296	160	140	350	319	334	289	406	335	376	286
487	364	408	303	154	147	360	329	341	296	395	339	374	291
448	354	396	295	156	134	338	332	335	295	396	338	386	303
468	354	391	304	163	145	348	313	341	298	393	342	391	291
543	370	413	311	152	148	415	309	329	295	409	356	372	277
655	372	401	303	150	141	366	368	381	292	414	352	389	301
578	366	400	315	161	137	350	310	369	298	424	312	377	282
531	353	394	307	167	142	351	296	332	303	412	348	395	296
478	357	412	307	154	139	346	309	346	298	399	347	384	300
473	368	412	303	162	146	340	300	347	297	415	337	377	285
462	368	409	301	163	141	346	300	333	282	408	345	389	314
484	361	419	307	156	141	349	308	334	304	407	350	378	283
479	350	395	311	155	146	338	302	336	295	423	342	389	309
447	360	408	297	160	141	352	312	341	302	432	335	419	287

Tabel 1

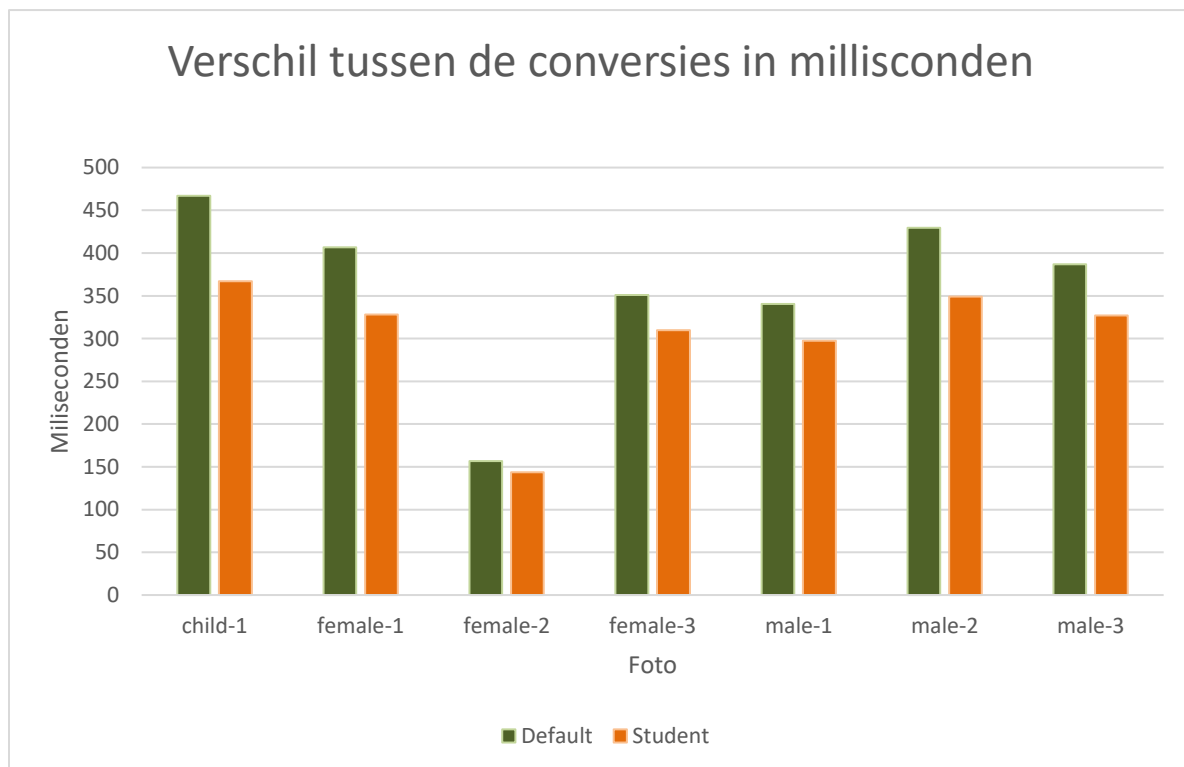
## Verwerking

Aan de hand van tabel 1 zijn er per kolom van de 50 conversies een gemiddelde berekend via de onderstaande formule:

$$f(x) = \frac{1}{n} \sum_{i=1}^n (x_i)$$

Voorbeeld bij child-1 default:  $\frac{1}{50} \cdot (939 + 729 + 631 + 508 + 567 \dots + 479 + 447) = 466,8$  milliseconden

Door dit bij alle kolommen uit te voeren, kwamen wij uit op de volgende grafiek:



Grafiek 1

Zoals in grafiek 1 te zien is, is onze conversie bij elke foto sneller. Het kleinste verschil zit bij female-2, de default conversie is 9% langzamer dan onze conversie. Het grootste verschil zit bij child-1. Hierbij is de default conversie 27% langzamer dan onze conversie.

## Conclusie

Uit de grafiek is gebleken dat onze conversie sneller is dan de default conversie. Dit snelheidsverschil is wel per foto afhankelijk. Onze hypothese klopte maar deels. Onze conversie is dus inderdaad sneller, maar niet 30% zo snel. Ons grootste verschil is maar 27% en de kleinste is 9%.

## Evaluatie

Zoals in de conclusie is geschreven, onze hypothese klopte maar deels. Er zijn natuurlijk veel meer stappen die het programma aflegt naast de conversieberekening. Hier hadden wij geen rekening mee gehouden.

Het doel van dit experiment was om te controleren of onze implementatie sneller is dan de default implementatie. Dit doel hebben wij goed kunnen bereiken.

Wij hebben tijdens dit experiment geen grote meetfouten kunnen ontdekken. Het enige wat misschien invloed heeft gehad, is dat wij eerst de default methode op een foto hebben toegepast, en toen pas onze student versie. Het kan zijn dat het ophalen van de foto op een ander adres de eerste paar keren in de default methode daarom iets langer duurde dan normaal. De andere keren kan het adres van de foto in de cash zijn opgeslagen, waardoor de executietijd minder lang zou duren.