TSE6-OpenCL: Sobel Edge Detect

De “Sobel operator” wordt veelal gebruikt in computer vision, voornamelijk voor edge detection. Om edges te herkennen heeft de Sobel filter twee “kernels” (klein matrix in image processing termen), in de X-richting en Y-richting. Berekeningen die hiermee worden uitgevoerd op de afbeelding berekenen dan de richting van de gedetecteerde edges.

Gezien elk van deze twee kernels onafhankelijk van een ander kan werken, kan het werk makkelijk worden opgedeeld over meerdere GPU-cores.

Bij deze opdracht heb ik het voorbeeld van Intel gepakt, en gecombineerd met mijn vorige opdrachten om een OpenCL-apparaat te kiezen. Hieraan heb ik nog mijn logger gehangen om voor elke aparte run een meting te doen.

Het voorbeeld van Intel verschillende optimalisatie-kernels:

1. Gebruik van uchar8. Veel compute threads!
2. Gebruik van uchar16. Minder compute threads ☺
3. Gebruik van uchar16 -> float16. GPU-ondersteuning – snellere berekeningen! ☺
4. 16x uchar16 in 1 kernel – elke GPU-core beter benut! ☺

Loop-unrolling kan worden toegepast in Kernel 4, maar dit is niet lonend voor kleinere afbeeldingen.

De Intel-GPU rapporteert een AMD-driver (?) dus het is niet gelukt metingen te doen met de interne GPU van de CPU.

Ik heb nog geprobeerd een ander bestand te gebruiken dan de voorbeeldhond, maar dit is niet gelukt wegens het .ppm (portable pixelmap) formaat, waar ik geen software voor kan vinden (Photoshop ondersteunt een net iets andere standaard).