

GEGEVENSSTRUCTUREN EN ALGORITMEN

Practicum 3: Image Compositing algoritme

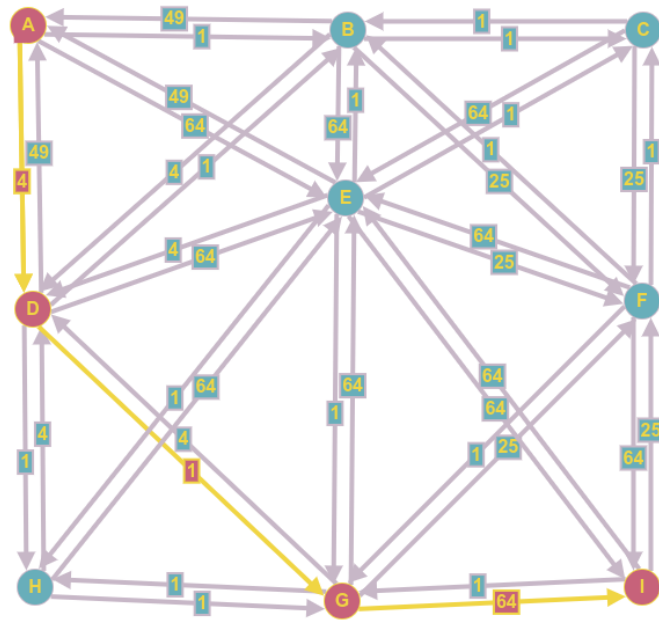
Author:

STEF TWEEPENNINCKX, R0677232

18 MEI 2018

Inleiding

Kortste pad



Figuur 1: Grafe met aangeduid kortste pad

Het resulterende kortste pad, aangeduid in het geel, is:

$$A \Rightarrow D \Rightarrow G \Rightarrow I$$

met een totale kost van $49 + 4 + 1 + 64 = 118$.

Andere afstandsfunctie

Als we de nieuwe afstandsfunctie, gegeven in de opgave, zouden gebruiken, houden we geen rekening meer met de blauwe component van een pixel. Aangezien we hierdoor minder berekeningen gaan doen, zal deze functie in de praktijk bij bijna iedere afbeelding een snellere uitvoeringstijd hebben.

Indien we gebruik maken van de oude afstandsfunctie, wordt eerst positie (0,1) gekozen. Daarna positie (1,0) en dan uiteindelijk eindpositie (1,1). Dijkstra heeft dus 3 stappen nodig om het einde te bereiken met de oude afstandsfunctie. Als we de nieuwe afstandsfunctie gebruiken zullen we rechtstreeks van (0,0) naar de eindpositie (1,1) gaan. In dit geval is er dus maar 1 stap nodig, wat duidelijk minder is en dus sneller.

(0,0,0)	(0,0,0)	(0,0,0)	(1,0,0)
(0,0,0)	(0,0,0)	(1,1,0)	(0,0,42)

Tabel 1: Twee images bij voorbeeld snellere uitvoeringstijd

Met de oude afstandsfunctie gaan we rechtstreeks van (0,0) naar eindpositie (1,1). De oude functie heeft dus 1 stap nodig om het einde te bereiken. Bij gebruik van de nieuwe afstandsfunctie zal Dijkstra eerst naar (0,1), dan naar (1,0) en dan pas naar eindpositie (1,1). In dit geval heeft de nieuwe functie 3 stappen nodig, meer dan de 1 stap van de oude functie en zal dus bijgevolg trager zijn dan de oude functie.

(0,0,0)	(0,0,0)	(0,0,0)	(0,0,45)
(0,0,0)	(0,0,0)	(1,0,42)	(1,1,0)

Tabel 2: Twee images bij voorbeeld trager uitvoeringstijd

Tijdscomplexiteit

Voorkomen van complexe vormen seam

Langste pad ipv kortste pad