Watershedgebaseerde segmentatie

Kristof Teelen

Labo beeldverwerking / 2007-08

1 Inleiding

Voor segmentatie van complexere beelden zijn de thresholding methodes van weinig nut, aangezien er meerdere gebieden met verschillende en/of gelijkaardige intensiteiten moeten onderscheiden worden. Er zijn betere technieken ontwikkeld om beelden te segmenteren. Een eerste interessante methode is de watershedtransformatie. Deze techniek is gebaseerd op een geografische interpretatie van een grijswaardenbeeld. In de geografie is een watershed is een richel die verschillende riviergebieden scheidt, en een cachment basin is de geografische regio waarin al het water in dezelfde rivier terechtkomt.

Grijswaardenbeelden worden voorgesteld als een topologisch gebied waarin alle objecten die in een beeld te zien zijn met een homogene intensiteitswaarde als een apart cachment basin beschouwd kunnen worden. De randen van die objecten moeten dan voorgesteld worden als watersheds. Een eenvoudige randdetectie is niet voldoende, aangezien randen objecten niet noodzakelijk volledig omsluiten (en meestal zelfs helemaal niet omsluiten, denk aan de randdetectoren in een vorige labo). Daarom zijn er andere technieken nodig die objecten als 1 aansluitend gebied kunnen voorstellen.

2 Watershed op basis van de afstandstransformatie

Een eerste methode om beelden te segmenteren in verschillende cachment basins is een watershedtransformatie op basis van de afstandstransformatie (distance transform). De afstandstransformatie van een binair (of logisch) beeld geeft voor elke pixel de afstand tot de dichtsbijzijnde achtergrondpixel. Dit vereist dus meestal eerst een thresholding stap. Een voorbeeld vind je hieronder (zie ook **help watershed** voor meer informatie) op het beeld **eurocoins.jpg**:

```
I = imread('images/eurocoins.jpg');
figure, imshow(I)
I = rgb2gray(I);
figure, imhist(I)
bw = im2bw(255-I,50/255);
figure, imshow(bw)
D = bwdist(~bw);
figure, imshow(D,[]), title('Distance transformatie van bw')
D = -D;
D(~bw) = -Inf;
L = watershed(D);
rgb = label2rgb(L,'jet',[.5 .5 .5]);
figure, imshow(rgb), title('Watershed transformatie van D')}
```

Het uitvoeren van deze stappen leidt tot het resultaat getoond in figuur 1:

3 Watershed op basis van de gradiënt-magnitude

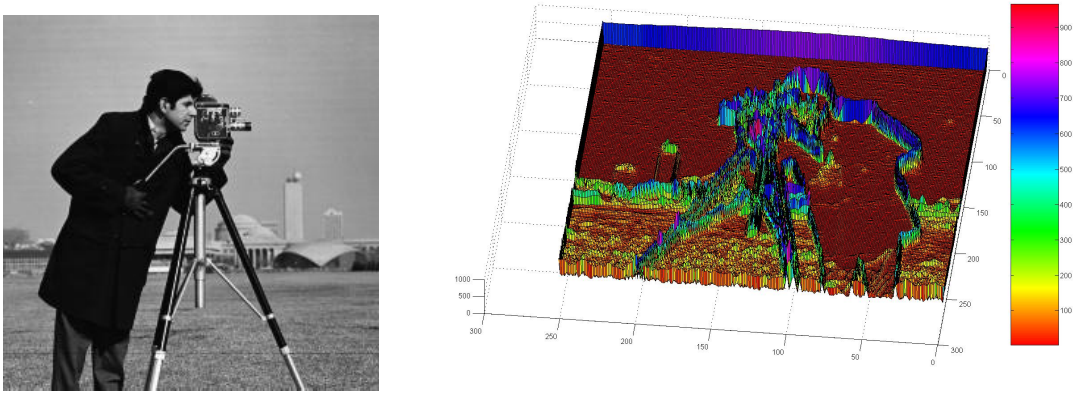
Een tweede concept dat je kan gebruiken om de watershed te berekenen zijn de gradiënten van een beeld: die zijn heel klein in regio's met constante intensiteit en heel hoog bij randen van objecten (die objecten

Figuur 1: Watershedtransformatie.

worden dan als de op te vullen watersheds beschouwd). Een vooorbeeld:

```
I = imread('images/cameraman.jpg');
f = [-1 -2 -1;0 0 0; 1 2 1];
imv1 = filter2(f, I);
imh1 = filter2(f',I);
gradmagn = sqrt(imv1.\^2+imh1.\^2);
figure, imshow(gradmagn,[]);
[X,Y] = meshgrid(1:size(I,1),1:size(I,2));
figure, surf(X,Y,gradmagn), colormap hsv, colorbar
```

Deze code berekend de magnitude van de gradiënt en toont het als een grijswaardenbeeld. In een tweede figuur wordt de gradiënt als een landschapsoppervlakte getoond, de grenzen van gebieden met een constante intensiteit zijn duidelijk zichtbaar in figuur 2.



Figuur 2: De gradiëntmagnitude van een beeld.

```
W = watershed(gradmagn);
figure, imshow(W)
```

Deze code resulteert in een binaire figuur waarin de pixels op (de randen van) de watersheds wit zijn en de regio's tussenin (= de catchment basins) zwart. Het resultaat is een overgesegmenteerd beeld: de gesegmenteerde gebieden komen op zich niet overeen met de objecten in het beeld, maar met onderdelen ervan. Dit is een veel voorkomend probleem bij dit soort algoritmes, en kan opgelost worden door achteraf gelijkaardige gebieden samen te voegen door region merging. Een gelijkaardig probleem is ondersegmentatie.

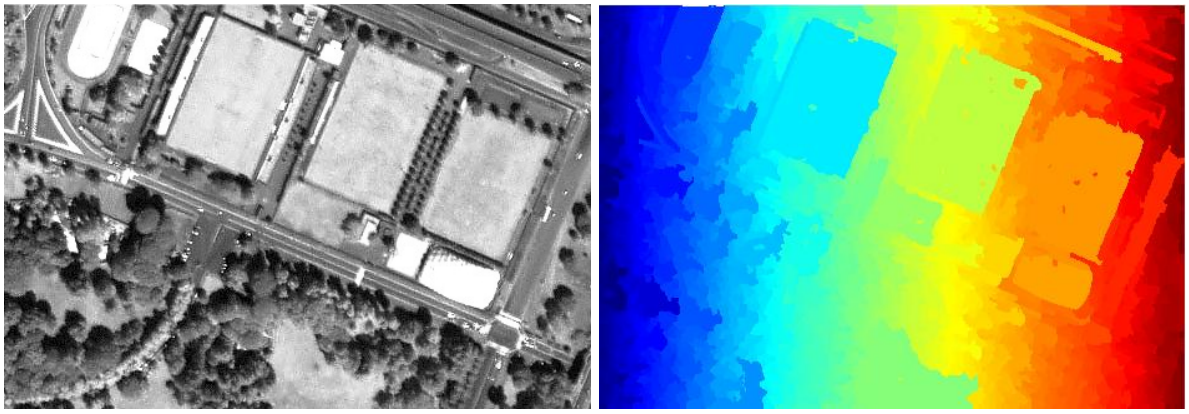
4 Flooding

Implementeer het volgende flooding-algoritme als een simpele variant op de watershed-transformatie. Bereken eerst ook een gradiënt-magnitudebeeld. De laagste intensiteitsgradiënten duiden op gebieden met een constante intensiteit. Deze regio's kan je dus gebruiken als initiële gebieden, die daarna stelselmatig uitgebreid

worden tot de eigenlijke cachment bassins. Vooral de aangrenzende pixels zijn hierbij van belang: je kan regio's enkel uitbreiden met pixels die grenzen aan de regio. Gebruik voor de uitbreiding als criterium dat de gradiënt mag niet te veel toegenomen zijn in vergelijking met de naburige pixel. Je kan gebruik maken van de code die al geschreven is in de file **flooding.m**. Zoek daarin de volgende stappen, en breidt uit indien nodig.

1. Bereken de initiële gebieden door een eerste drempelwaarde T_G op de waarde van de gradi¨ent toe te passen en label de geconnecterde gebieden m.b.v. **imlabel**.
2. Verhoog de drempelwaarde T_G met een constante waarde A_{T_G}.
3. Zoek de aangrenzende pixels van de reeds gelabelde gebieden. Indien deze pixels een gradiëntwaarde onder de huidige waarde voor T_G hebben, voeg je ze toe aan het aangrenzende gebied door aan die pixels hetzelfde label toe te kennen.
4. Herhaal stap 2 en 3 tot alle pixels gelabeld zijn.

Pas het algoritme toe op het beeld **satim.jpg**. Deze segmentatie kan dienen als eerste stap in een herkenningsalgoritme voor velden, wegen, gebouwen, ... op satelietfoto's. Wat zijn de voor- en nadelen van deze methode?



Figuur 3: Het resultaat van het beschreven floodingalgoritme voor satim.jpg.

5 Rainfalling

Er bestaan nog andere segmentatie-algoritmes zoals rainfalling, waarbij in elke pixel aangeduidt wordt in welke richting een regendruppel zou vloeien als die daar zou vallen (dus in de richting van de sterkst dalende gradiënt). Op die manier kan je de paden langs dalende gradiënten volgen tot in een lokaal minimum van de gradiëntmagnitude en zo de catchment basins definiëren. Wat is het voordeel van deze methode ten opzicht van de floodingmethode?