Special Lecture for Computer Science – COSE490

Digital Image Processing :: Level-set Image Segmentation

Due date : 2022-11-27

고려대학교 컴퓨터학과 2017320108

고재영

개발 환경 : Matlab Desktop

과제 만기일 : 2022-11-27

제출 날짜 : 2022-11-20

최종 제출: 2022-11-20

재제출 사유: none

[0] – 과제 설명

- In this assignment, you will implement a level-set image segmentation method. The algorithm we are implementing is Geodesic Active Contour formulation of level set method by Caselles et al.

- The provided assign_3_skeleton.m is the main driver code that calls the level set update function (levelset_update.m). In the main driver code, you can modify the parameter values (dt, c, niter, k) and the following part to compute the edge indicator term g.

- Run the experiment with various combination of dt and c on different testing images and discuss the result in the report. Submit the report (pdf) and source code via blackboard.

[1] – levelset_update.m

첫 번째로 이 'levelset_update.m' 함수는 phi에 관한 distance field를 받아 해당 값을 업데이트 시켜주는 역할을 수행하는 함수이다.

해당 함수의 코드는 다음과 같다.

```
%
% Skeleton code for COSE490 Fall 2022 Assignment 3
%
% Won-Ki Jeong (wkjeong@korea.ac.kr)
%

function phi_out = levelset_update(phi_in, g, c, timestep)
phi_out = phi_in;

%
% ToDo
%

% inputs ::
% g as edge term
% c as constant term
% timestep as time derivative

% gradient
[phi_x, phi_y] = gradient(phi_in);
% for normalization
mag_gradPhi = sqrt(phi_x.^2 + phi_y.^2);
% to avoid zero-dividing, add little possitive value
lil_pos = 0.0000000001;

dphix = phi_x./(mag_gradPhi + lil_pos);
dphiy = phi_y./(mag_gradPhi + lil_pos);

[dphix2, dphixy] = gradient(dphix);
[dphiyx, dphiy2]= gradient(dphiy);
div_phi = dphix2 + dphiy2;

% <1> First compute 'dPhi'
dPhi = mag_gradPhi; % mag(grad(phi))

%
% <2> Second, compute kappa
kappa = div_phi; % curvature

%%%

smoothness = g.*kappa.*dPhi;
expand = c*g.*dPhi;
```

```
phi_out = phi_out + timestep*(expand + smoothness);
```

Phi_in, g, c, timestep 으로 총 네 가지의 parameter를 받아서, 입력받은 phi_in 값에 대해 expand와 smoothness의 합을 timestep에 곱해준 값과 더한 phi_out을 출력한다. 이 때, expand와 smoothness는 각각 contour 이동에 관한 constant term 'c'와 smoothness 개선을 위한 curvature term 'k'와 관련한다.

Phi_in을 통해 gradient를 계산하며, 크기로 나누어주어 normalization을 해준다. 이 때, 한 가지 주의할 점은 0으로 나눌 수 없다는 점을 간과할 수 있기 때문에 필자는 해당 발생가능한 문제를 회피하기 위해 'lil_pos' 라는 매우 작은 positive numerical value를 더해주는 방식을 사용했다.

Curvature term인 'k' (코드 내에서 kappa 변수) 는 second-derivative term으로 볼 수 있기에 위의 코드처럼 값을 도출하도록 구현했다.

[2] – assign_3_skeleton.m

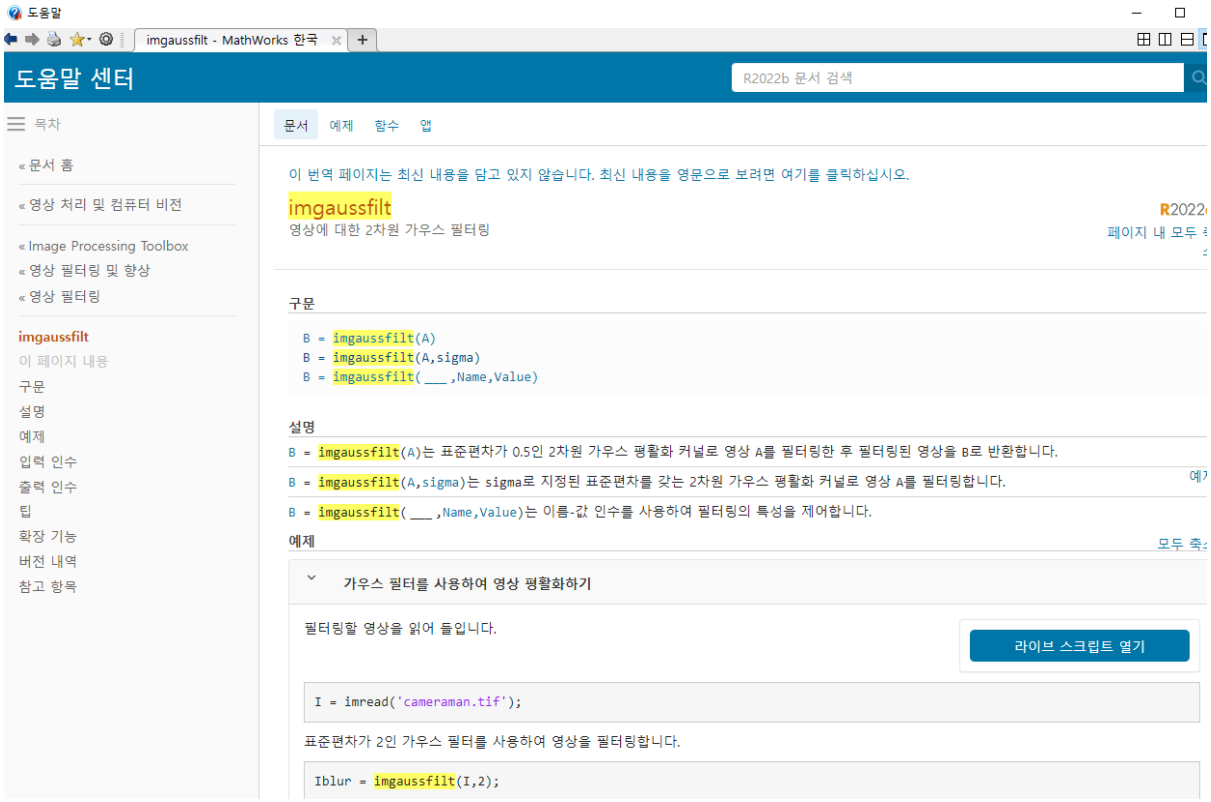
위에서 소개한 'levelset_update.m' 함수를 직접 사용하는, 메인 드라이버 코드 부분에 해당한다. 여러 parameter 값을 조정하면서 이미지 분할의 과정 결과를 확인해 볼 수 있다.

해당 함수의 코드는 마찬가지로 다음과 같다.

```
%
% Skeleton code for COSE490 Fall 2022 Assignment 3
%
% Won-Ki Jeong (wkjeong@korea.ac.kr)
%

clear all;
close all;

%
% Loading input image
%
Img=imread('coins-small.bmp');
Img=double(Img(:,:,1));

%
% Parameter setting - modify as you wish
%
dt = 0.8;  % time step
c = 1.0;  % weight for expanding term
niter = 400; % max # of iterations


%
% Initializing distance field phi
%
% Inner region : -2, Outer region : +2, Contour : 0
%
[numRows,numCols] = size(Img);
phi=2*ones(size(Img));
phi(10:numRows-10, 10:numCols-10)=-2;

%
% Compute g (edge indicator, computed only once)
%

% ToDO ------------------------

% using gaussian smoothing
% p norm as L-2 norm (square)

% <1> gaussian smoothing
% in matlab function imgaussfilt, we can moderate sigma value
sigma = 2;
img_smoothed = imgaussfilt(Img, sigma);

% <2> get g from smoothed image
```

```
[img_dx, img_dy] = gradient(img_smoothed);
grad_ihat = sqrt(img_dx.^2 + img_dy.^2);

g = 1 ./ (1 + grad_ihat.^2);


% ----------------------------

%
% Level set iteration
%
for n=1:niter

    %
    % Level set update function
    %
    phi = levelset_update(phi, g, c, dt);

    %
    % Display current level set once every k iterations
        %
        % Modify k to adjust the refresh rate of the viewer
    %
    k = 10;
    if mod(n,k)==0
        figure(1);
        imagesc(Img,[0, 255]); axis off; axis equal; colormap(gray); hold on;
contour(phi, [0,0], 'r');
        str=['Iteration : ', num2str(n)];
        title(str);

    end
end


%
% Output result
%
figure(1);
imagesc(Img,[0, 255]); axis off; axis equal; colormap(gray); hold on;
contour(phi, [0,0], 'r');
str=['Final level set after ', num2str(niter), ' iterations'];
title(str);
```

위 코드에서 핵심적인 부분은 edge indicator term에 관한, g 구현에 관한 것이다. 이에 앞서서, input image에 대해서 smoothing의 전처리 과정을 적용해야 한다. 필자는 Gaussian Smoothing 필터를 이용한 방식을 채택했다. 일반적으로 사용되는 방법이기도 하며, 무엇보다 matlab의 내장함수 imgaussfilt를 이용하는 게 편하기 때문이다.



해당 함수는 특히 sigma 값으로 표준편차를 지정하여 gaussian filtering을 조절할 수 있다. 이 방법으로 smoothing 전처리를 마친 이미지에 대하여, L-2 norm (p = 2, 제곱 형태)으로 적용하여 g를 위 코드와 같이 계산할 수 있다.

[3] – Discussion for various cases

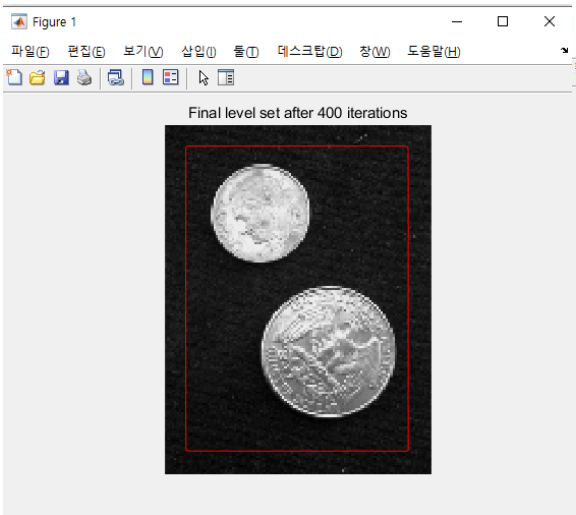
완성한 코드로부터 여러 가지 parameter값에 대해 변화를 주며 결과를 관찰한 결과에 대한 설명이다. 필자가 통제할 수 있는 변수는 다음과 같다.

```
dt = 0.8;  % time step
c = 1.0;  % weight for expanding term
niter = 400; % max # of iterations


%
    % Display current level set once every k iterations
        %
        % Modify k to adjust the refresh rate of the viewer
    %
    k = 10;
```

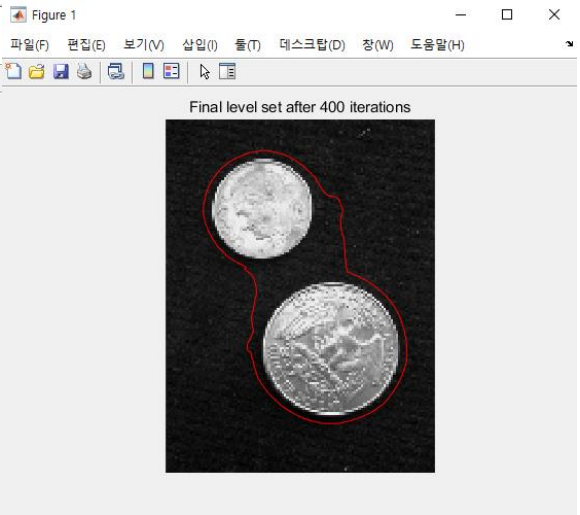
변인 통제를 위해, 위 네 변수에 대해 skeleton에서 주어졌던 값을 default 기준으로 잡았다.

<1> dt

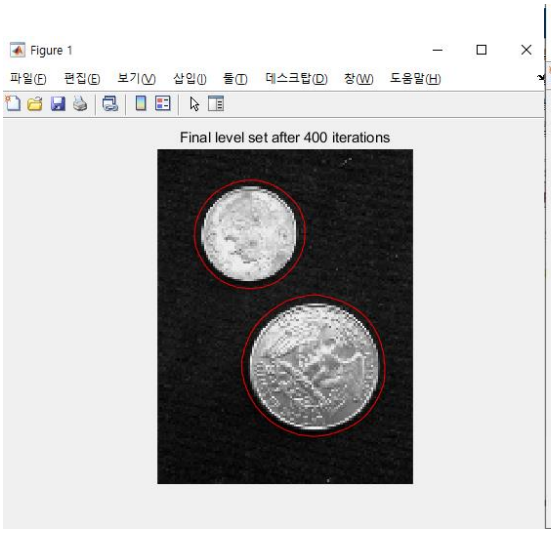
먼저 dt에 대해 살펴보자. Time step 내지는 Deep Learning 분야와 관련해서 보통 learning rate (step size)로 불리는 dt의 값에 대한 결과이다.



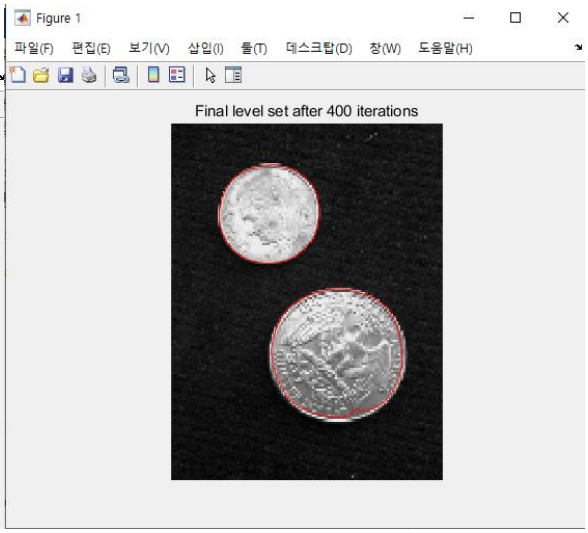
Dt = 0.001



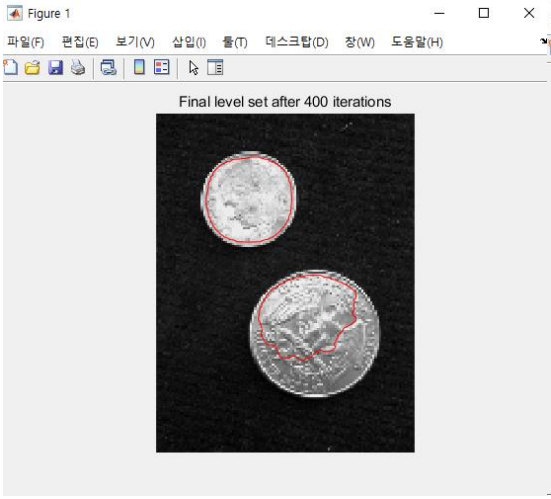
dt = 0.1



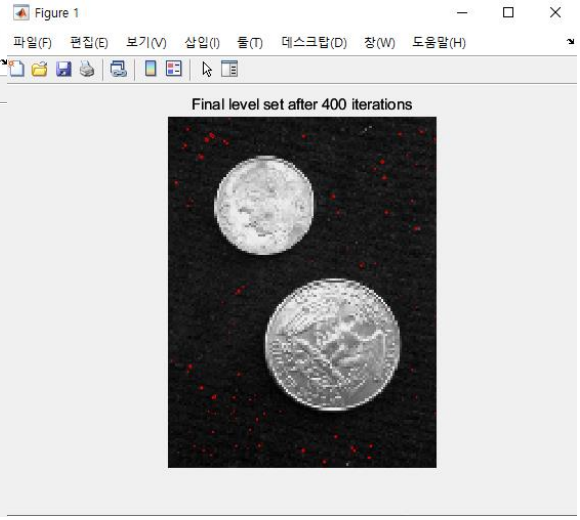
Dt = 0.2



dt = 0.8



Dt = 1



dt = 2

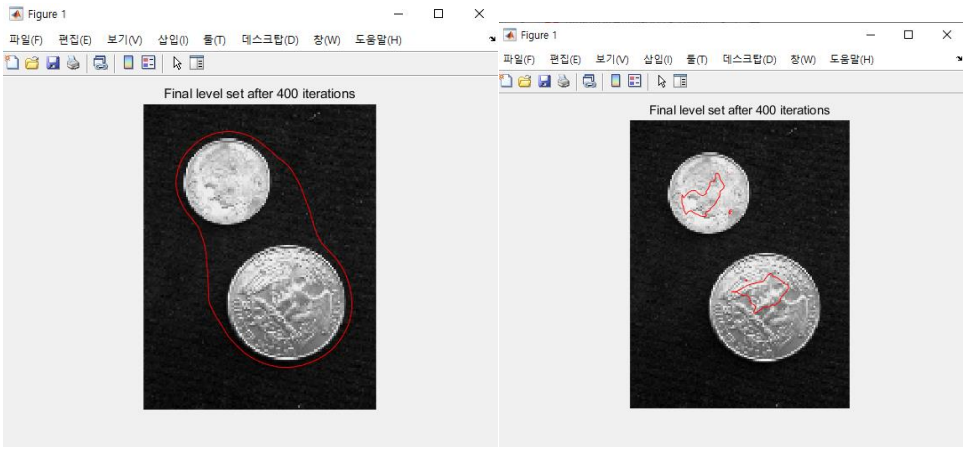
Dt가 너무 작을 경우, iteration이 반복되더라도 distance field가 수렴할 때까지 줄어들지 못하는 것을 dt = 0.001과 0.1일 때를 통해 볼 수 있다. 마치 machine learning에서 너무 작은 learning rate으로 underfitting이 일어난 것과 비슷하게 생각할 수 있다.

Dt = 0.2, 0.8 일 땐 보통의 목적에 부합한 이미지 분할이 잘 일어난 것을 볼 수 있다.

Dt = 1, 2일 때는 learning rate이 과도하게 커서 overshooting되는 결과를 볼 수 있다.

<2> c

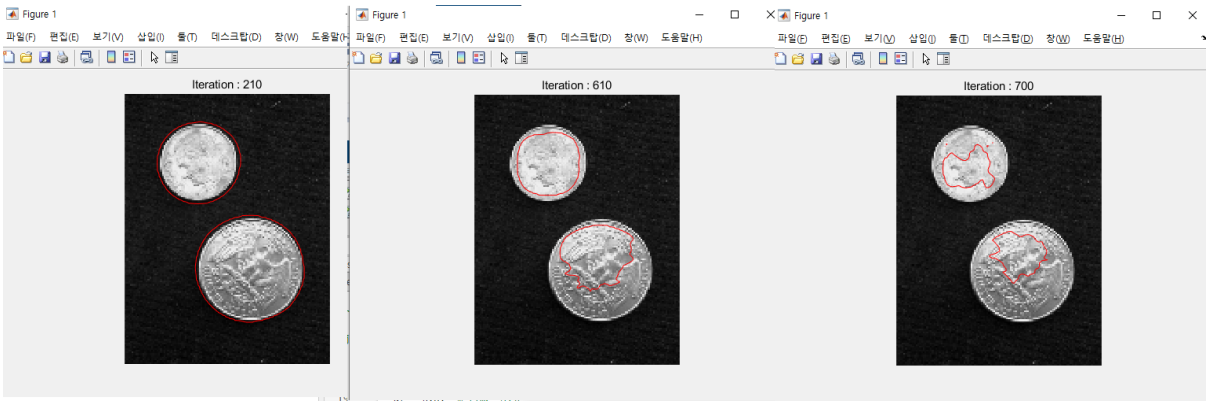
Expanding term으로서 c



C = 0.1 c = 1.5

Expanding의 정도에 영향을 미치는 것을 결과를 통해 직관적으로 알아볼 수 있다.

<3> number of iterations



Iteration을 1000으로 설정하고 특정 시점마다 스크린샷을 찍어 얻은 결과이다.

<4> k

k값은 사실 결과를 측정하는 view에게 보여지는 수렴 과정 중 시점에 관한 것이기 때문에 k를 건드리는 것은 이미지 분할 결과에 영향을 미치지 않는다.