

function [x, y, bw] = getMouth(pic)

%

% Purpose : find position of Mouth within an Image

%

% Uses (syntax) :

% [x, y, bw] = getMouth(pic)

%

% Input Parameters :

% pic := RGB-Image (m-by-n-by-3 matrix); bounding box of Mouth

%

% Return Parameters :

% x := x-value of Eye's center

% y := y-value of Eye's center

% bw := Binary-Image (m-by-n matrix)

%

% Description and algorithms:

% Calculates the position of the Mouth based on the Image's Hue and

% Saturation

%

% Author : Peter Aldrian, Uwe Meier, Andre Pura

% Date : August 12, 2009

% Version : 1.0

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

% (c) 2009, Meier, University of Leoben, Leoben, Austria

% email: aldrian.peter@gmail.com uwemei@gmail.com andre.pura@gmail.com

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

si = size(pic);

% Convert RGB to HSV

[h,s,v] = rgb2hsv(pic);

% Erode Hue

se = strel('disk',3);

erodedHue = imerode(h,se);

% Dilate Hue

se = strel('disk',3);

dilatedErodedHue = imdilate(erodedHue,se);

% Convert Image to binary Image

bw = im2bw(dilatedErodedHue,graythresh(dilatedErodedHue));

% multiply found values with saturation and only use pixel which have a

% value above a certain thresh

s = s .\* bw;

maximum = max(max(s));

s(s<maximum\*0.5)=0;

% Find non-zero pixels and calculate mean point

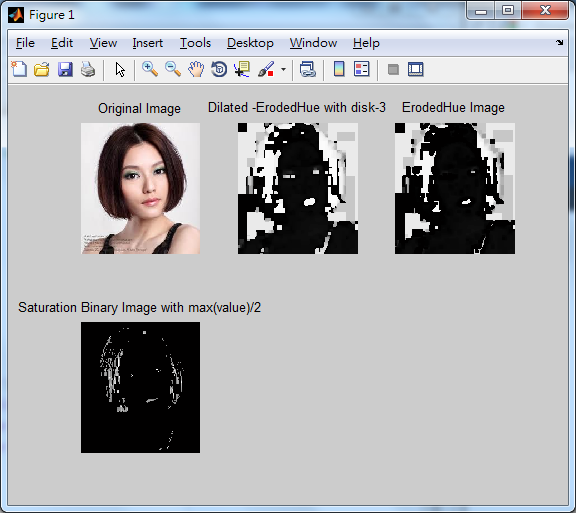
[y x] = find(s);

x = mean(x);

y = mean(y);

clearvars -except x y bw

end



function [x, y, bw] = getMouth(pic)

%

% Purpose : find position of Mouth within an Image

%

% Uses (syntax) :

% [x, y, bw] = getMouth(pic)

%

% Input Parameters :

% pic := RGB-Image (m-by-n-by-3 matrix); bounding box of Mouth

%

% Return Parameters :

% x := x-value of Eye's center

% y := y-value of Eye's center

% bw := Binary-Image (m-by-n matrix)

%

% Description and algorithms:

% Calculates the position of the Mouth based on the Image's Hue and

% Saturation

%

% Author : Peter Aldrian, Uwe Meier, Andre Pura

% Date : August 12, 2009

% Version : 1.0

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

% (c) 2009, Meier, University of Leoben, Leoben, Austria

% email: aldrian.peter@gmail.com uwemei@gmail.com andre.pura@gmail.com

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

si = size(pic);

% Convert RGB to HSV

[h,s,v] = rgb2hsv(pic);

% Erode Hue

se = strel('disk',3);

erodedHue = imerode(h,se);

% Dilate Hue

se = strel('disk',3);

dilatedErodedHue = imdilate(erodedHue,se);

% Convert Image to binary Image

bw = im2bw(dilatedErodedHue,graythresh(dilatedErodedHue));

% multiply found values with saturation and only use pixel which have a

% value above a certain thresh

s = s .\* bw;

maximum = max(max(s));

s(s<maximum\*0.5)=0;

% Find non-zero pixels and calculate mean point

[y x] = find(s);

x = mean(x);

y = mean(y);

subplot(2,3,1),imshow(pic);title('Original Image')

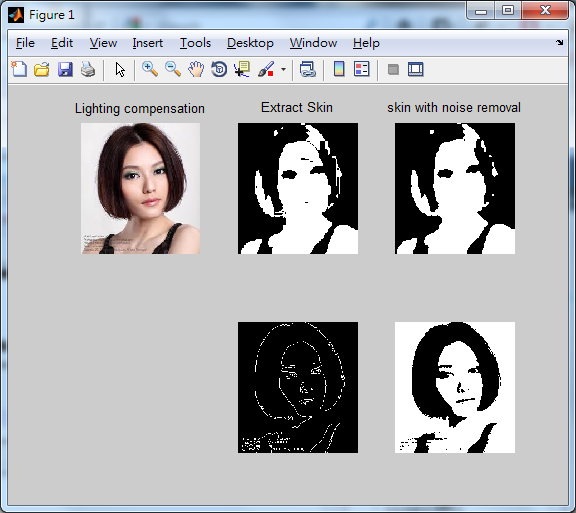
subplot(2,3,2),imshow(dilatedErodedHue);title('Dilated -ErodedHue with disk-3')

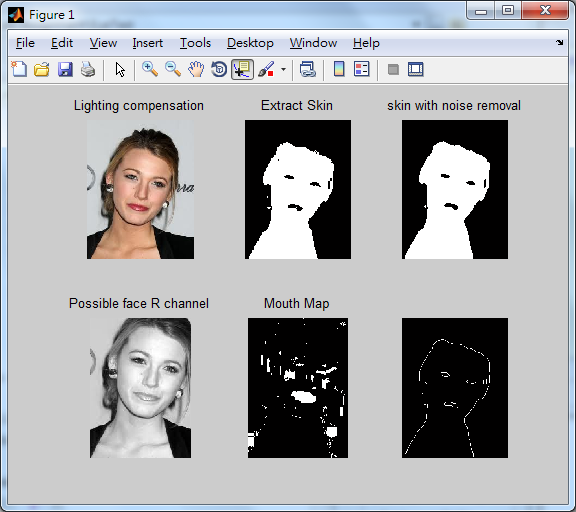
subplot(2,3,3),imshow(erodedHue);title('ErodedHue Image')

subplot(2,3,4),imshow(s);title('Saturation Binary Image with max(value)/2');

clearvars -except x y bw

end





ISN = ~SN;

imshow(ISN)

K>> radii = 5:1:15;

K>> e = edge(ISN, 'canny');h = circle\_hough(e, radii, 'same', 'normalise');

imshow(ISN)

CC=bwconncomp(ISN,4)

CC =

Connectivity: 4

ImageSize: [257 196]

NumObjects: 5

PixelIdxList: {1x5 cell}

ISN = ~SN

imshow(ISN)

SE2=ones(257\*196)

CC=bwconncomp(ISN,4)

S = regionprops(CC,'Area')

L = labelmatrix(CC)

Eyemin = round(W/20 \* H/30)

Eyemax = round(W/10 \*H/15)

BW3 = ismember(L,find([S.Area] >= Eyemin))

BW4 = ismember(L,find([S.Area] <= Eyemax)

subplot(1,2,1); imshow(BW3);

subplot(1,2,2);imshow(BW4);

