

Fuzzy sets. Operations with fuzzy sets applied to colour image segmentation.

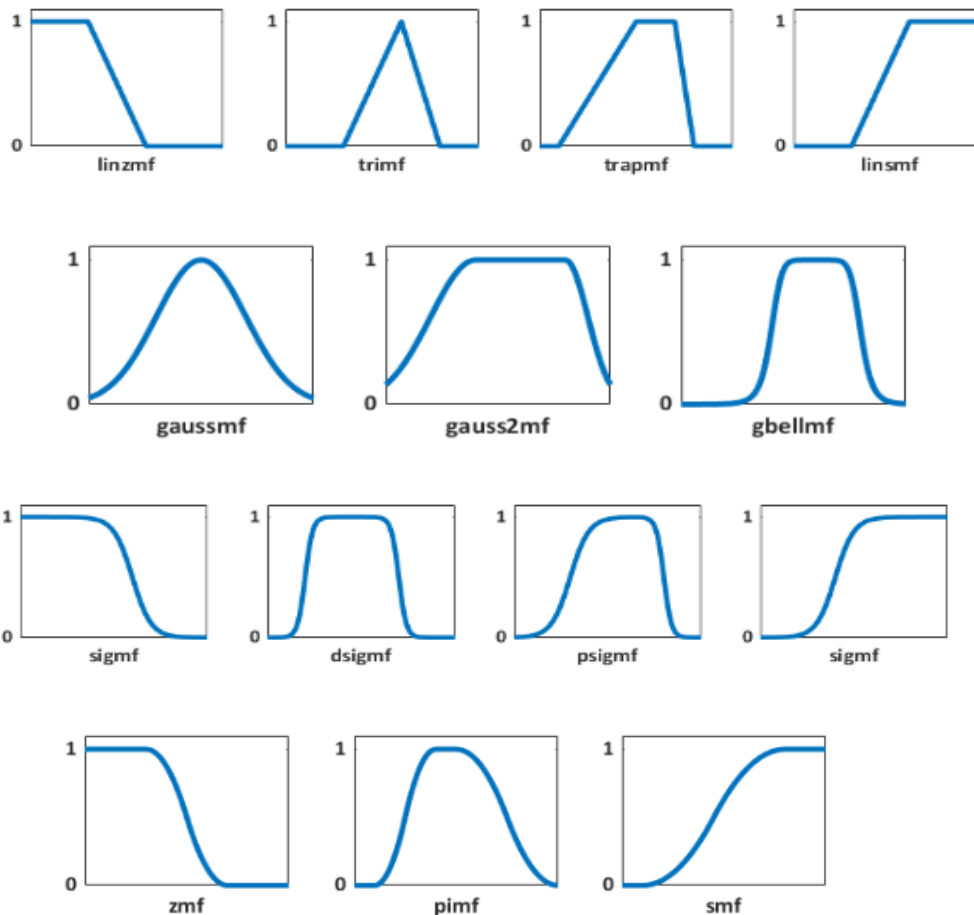
Objectives: getting used to the functions for defining fuzzy sets, understanding basic fuzzy sets operations, understanding the way colours are represented in MATLAB, understanding the basic operations for image processing and image segmentation

Note: MATLAB/Simulink is accessed online (<https://matlab.mathworks.com/>), by logging in with the MS Teams student credentials (surname.name@student.utcluj.ro).

Terms and abbreviations: *fuzzy set, membership function, union, intersection, complement, segmentation*

- Types of membership functions. Using and editing a membership function.

MATLAB offers 13 types of membership functions (MF – membership function):



To view details about any function, as well as a brief usage example, use “*help function_name*”:

```
help trapmf
```

Exercise 1

Copy and paste the following code snippet in the command line. Analyse the code, line by line.

```
close all % closes all the open figure windows
clear all % removes all variables in the workspace
clc % clears the command window
x = (0:0.1:10)'; % The universe of discourse is [0,10]; the points are
%defined with a step of 0.1
params=[2 3 7 9];% parameters for a trapezoidal membership function
y = trapmf(x,params); % compute the membership function values
plot(x, y, 'linewidth',2);
axis([0 10 -0.1 1.1]);
xlabel('universe of discourse'); % horizontal axis variable name
ylabel ('membership degree'); % vertical axis variable name
set(gcf, 'name', 'Trapezoidal membership function', 'numbertitle', 'off');
% figure name
```

Exercise 2

Run the next sequence to find the membership degree of the point $x_1=2.75$, to the above defined trapezoidal fuzzy set.

```
hold on
x1=2.75;
u1=evalmf(x1,params,'trapmf');
sprintf('x1=%1.2f has the membership degree u1=%1.2f',x1,u1);
plot(x1,u1,'r*') % place the point on the graph
plot([x1,x1],[0,u1],...
'linestyle','-','color','r')
plot([0,x1],[u1,u1],...
'linestyle','-','color','r')
hold off
```

Exercise 3

Let “Speed” be a linguistic variable, with the universe of discourse [0, 140] km/h. Define five linguistic values and plot the corresponding membership functions on the same axis (using “*hold on*”). The fuzzy sets must form a fuzzy partition for the [0,140] range. Display at the command line the values of the membership degrees of points 10 km/h, 52 km/h, 85 km/h and 100 km/h to each fuzzy set.

○ Fuzzy sets operations

The union of two fuzzy sets A and B is defined using the membership functions:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

The intersection of two fuzzy sets A and B is defined as:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

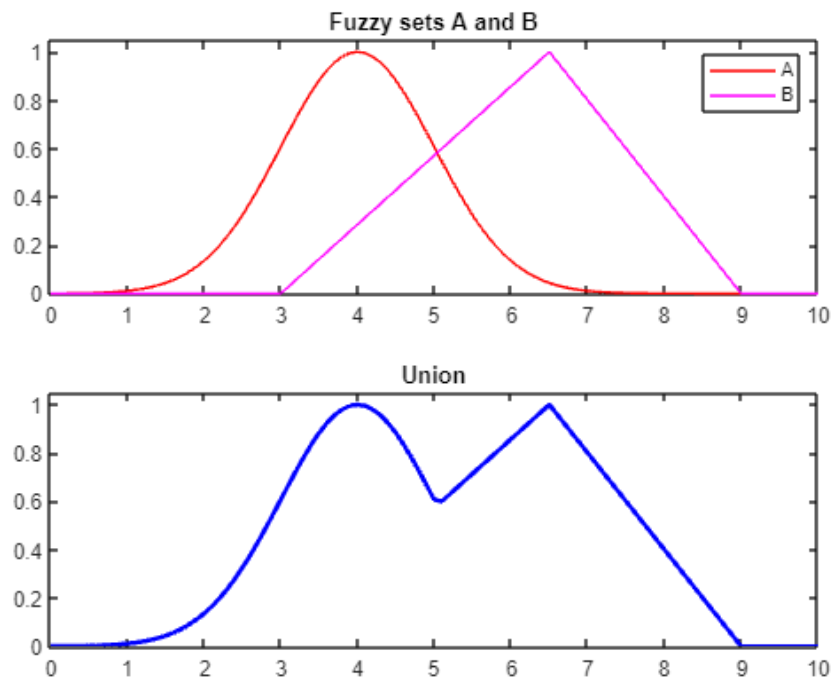
The complement of a fuzzy set A is defined as:

$$\overline{\mu}_A(x) = 1 - \mu_A(x)$$

Exercise 4

Run the following code snippet, to plot the union of two fuzzy sets. Analyse the code, line by line.

```
close all
x = (0:0.1:10)'; % % The universe of discourse is [0,10]; the points are
%defined with a step of 0.1
u1=gaussmf(x,[1,4]); % first membership function, gaussian type
u2=trimf(x,[3 6.5 9]); % second membership function, triangular type
u_union=max(u1,u2); % compute the membership degrees for the union using
%the "MAX" operator
hold on
subplot(2,1,1); % breaks the figure in two windows
% the current plot appears in the upper window
plot(x,u1,'r');hold on
plot(x,u2,'m'); hold off
axis([0 10 0 1.05]);
legend('A','B');
title('Fuzzy sets A and B')
subplot(2,1,2) % the current plot appears in the lower window
plot(x, u_union,'color','b','linewidth',2)
axis([0 10 0 1.05]);
title('Union')
set(gcf, 'name', 'Union of A and B fuzzy sets - "max" operator',
'numbertitle', 'off'); %figure name
```



Exercise 5

Change the previous code snippet, to plot in a single figure, on different subplots, the following: fuzzy sets A and B, union, intersection, complement.

- **Colour representation in MATLAB. RGB and HSV colour spaces.**

The **RGB** (red, green, blue) colour space is defined by the three primary colours, red, green and blue, out of which any other colour can be obtained, by weighted additive combinations. The values of R, G and B are in [0;1]. To represent any colour in MATLAB in the RGB space, one can use either the name (full or abbreviated) of the colour, for predefined colours, or a vector that contains the values of the primary components.

To set a pink background for the current figure, use the following code in the command line:

```
figure
hold on
set(gcf, 'Color', [1,0.4,0.6]);
```

Exercise 6

Change the background colour of the current figure to cyan, then to green, using both the abbreviated name and the vector representation.

In the **HSV** (hue, saturation, value) colour space, colours are represented by the perceptual attributes – hue, saturation, and value. The values for H, S and V are in [0;1]. The default colour representation in MATLAB is in the RGB space. To convert from RGB to HSV, the dedicated function is "rgb2hsv".

- **Representation of colours using fuzzy sets**

The values between 0 and 1 of the hue component, H, define the colours of the visible spectrum: red, orange, yellow, green, cyan, blue, purple, magenta, pink, and red again, at the end of the spectrum. To represent colours as fuzzy sets, the universe of discourse is the range of possible values for hue; when using an 8-bit representation, the range of values for H is [0; 255]. A different fuzzy set is defined for each colour, and the linguistic values are: Red, Orange, Yellow, Green, Cyan, Blue, Purple, Magenta, Pink, over the linguistic variable H (range [0; 255]), with values in [0; 1].

Exercise 7

Run the following code snippet. Analyse the code, line by line.

```
close all;
clear all;
clc;

% fuzzy sets definition
x=(0:.5:255)'; % universe of discourse
Red_l=trimf(x, [0 0 21]); % red, left
Red_r=trimf(x, [234 255 255]); % red, right
Red=max(Red_l, Red_r);
Orange=trimf(x, [0 21 43]); % orange
Yellow=trimf(x, [21 43 80]); % yellow
Green=trapmf(x, [43 80 90 128]); % green
Cyan=trimf(x, [90 128 165]); % cyan
```

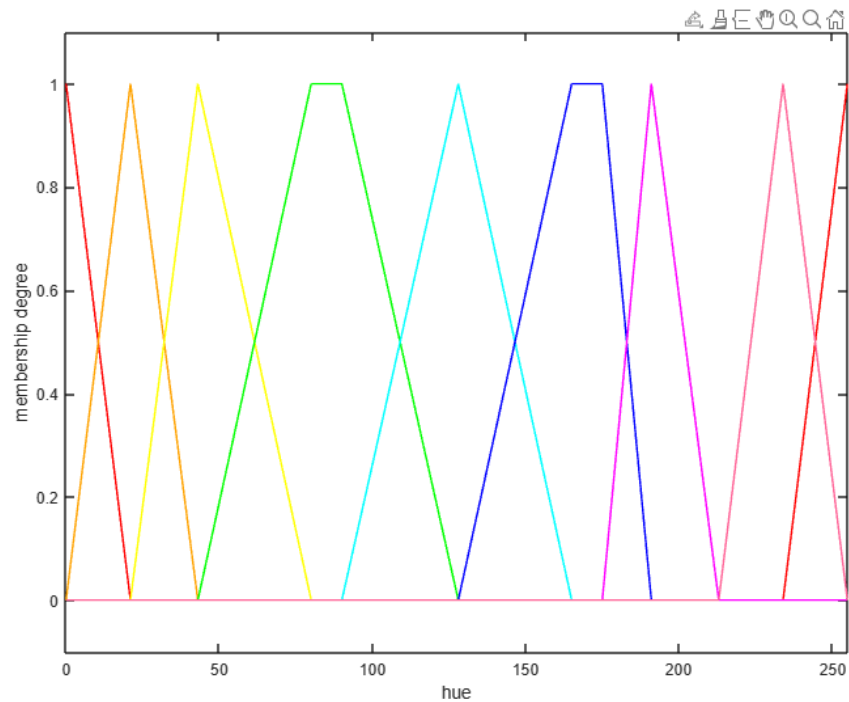
```

Blue=trapmf(x, [128 165 175 191]); % blue
Purple=trimf(x, [175 191 213]); % purple
Pink=trimf(x, [213 234 255]); % pink

plot(x, Red_l, 'color', 'r');
axis ([0 255 -.1 1.1]);
hold on;
xlabel('hue'); % x axis variable name
ylabel ('membership degree'); % y axis variable name
set(gcf, 'name', 'Representation of colours as fuzzy sets', 'numbertitle',
'off'); % figure name

plot(x, Red_r, 'color', 'r');
hold on;
plot(x, Orange, 'color', [1 0.64 0]);
hold on;
plot(x, Yellow, 'color', 'y');
hold on;
plot(x, Green, 'color', 'g');
hold on;
plot(x, Cyan, 'color', 'c');
hold on;
plot(x, Blue, 'color', 'b');
hold on;
plot(x, Purple, 'color', 'm' );
hold on;
plot(x, Pink, 'color', [1,0.4,0.6]);
hold off;

```



- Segmentation of a colour image

Segmentation of a colour image means isolating areas of a certain colour in an image. Let's assume the red colour segmentation is desired: the areas or objects coloured in red will be separated, by representing them using shades of grey. The shades of grey will be very close to white, if the object is almost red, and darker (even black), if the object is any other colour.

Using the previously defined fuzzy sets for colours, a new fuzzy set is created, called *CloseToRed*, that expresses the concept of objects with red dominant colour or objects that are very close to red. The *CloseToRed* fuzzy set is defined using the basic operations:

$$\text{CloseToRed} = \text{Red OR (Orange NOT Yellow) OR Pink}$$

$$\text{CloseToRed}(H) = \max(\text{Red}(H), \min(\text{Orange}(H), 1 - \text{Yellow}(H)), \text{Pink}(H))$$

Exercise 8

Run the following code snippet to plot the *CloseToRed* fuzzy set. Analyse the code, line by line.

```
figure;
NOT_Yellow=1-Yellow;
OrangeNotYellow=min(Orange, NOT_Yellow);
CloseRed=max(Red, OrangeNotYellow);
CloseToRed=max(CloseRed, Pink);
plot(x, CloseToRed, 'Linewidth', 2, 'color','r');
axis ([0 255 -1 1.1]);
xlabel('hue'); % x axis variable name
ylabel('membership degree'); % y axis variable name
set(gcf, 'name', 'CloseToRed fuzzy set', 'numbertitle', 'off'); % figure
%name
```

The membership degree of a pixel to the *CloseToRed* fuzzy set is proportional to its shade of red. The visual representation of the result is done by scaling the resulting membership degree in the [0;255], for each pixel. In the segmented image, the red or almost red areas will be replaced by white or almost white areas.

Exercise 9

Download "*images.zip*" and place the archive (using *drag-and-drop*) in the current directory of MATLAB. Double click to unzip and view the contents of the folder.

<http://www.bel.utcluj.ro/dce/didactic/sf/lab/3OperatiiMFSegmentareColor/images.zip>

Run the following code snippet to segment a colour image. Analyse the result of the segmentation for each image in the folder.

```
close all; clc; clear all
% available images; use only one at once
img='Tiffany24.bmp';
% img='Cuticle.bmp';
% img='Lenna.bmp';
% img='Mouse.bmp';
% img='Peppers2.bmp';
% img='Boats24.bmp';

OrigImg=imread(img); % reading the image
```

```

HSVimg=(rgb2hsv(OrigImg)); % RGB->HSV conversion
Himg=(HSVimg(:,:,1)); % extracting the hue component, between [0,1]
H=(Himg*255); % normalizing to [0, 255]

% computation of the membership degree of each pixel, to each fuzzy set
RedH_l=trimf(H, [0 0 21]);
RedH_r=trimf(H, [234 255 255]);
RedH=max(RedH_l, RedH_r);
OrangeH=trimf(H, [0 21 43]);
YellowH=trimf(H, [21 43 80]);
GreenH=trapmf(H, [43 80 90 128]);
CyanH=trimf(H, [90 128 165]);
BlueH=trapmf(H, [128 165 175 191]);
PurpleH=trimf(H, [175 191 213]);
PinkH=trimf(H, [213 234 255]);

% normalizing to [0, 255]
RedObj=uint8(255*RedH);
% creating the "CloseToRed" fuzzy set
NOT_YellowH=1-YellowH;
OrangeNotYellowH=min(OrangeH, NOT_YellowH);
CloseRedH=max(RedH, OrangeNotYellowH);
CloseToRedH=max(CloseRedH, PinkH);
% displaying the images
imshow(OrigImg); set(gcf, 'name', 'Original image'); % original image
figure
imshow(RedObj); set(gcf, 'name', 'Red segmentation'); % highlighting the red
%segments
figure
imshow(uint8(CloseToRedH*255)); set(gcf, 'name', 'Close to red
segmentation'); % highlighting the close to red segments

```



Original image



Red segmentation



Close to red segmentation

Exercise 10

Create a script to define a fuzzy set that approximates a different colour (other than red). Plot the fuzzy set. Apply the segmentation using the pure colour and the approximated colour.

Use the images in the folder to test the segmentation or download other relevant images.