

Mulțimi fuzzy. Operații cu mulțimi fuzzy aplicate în segmentarea imaginilor color.

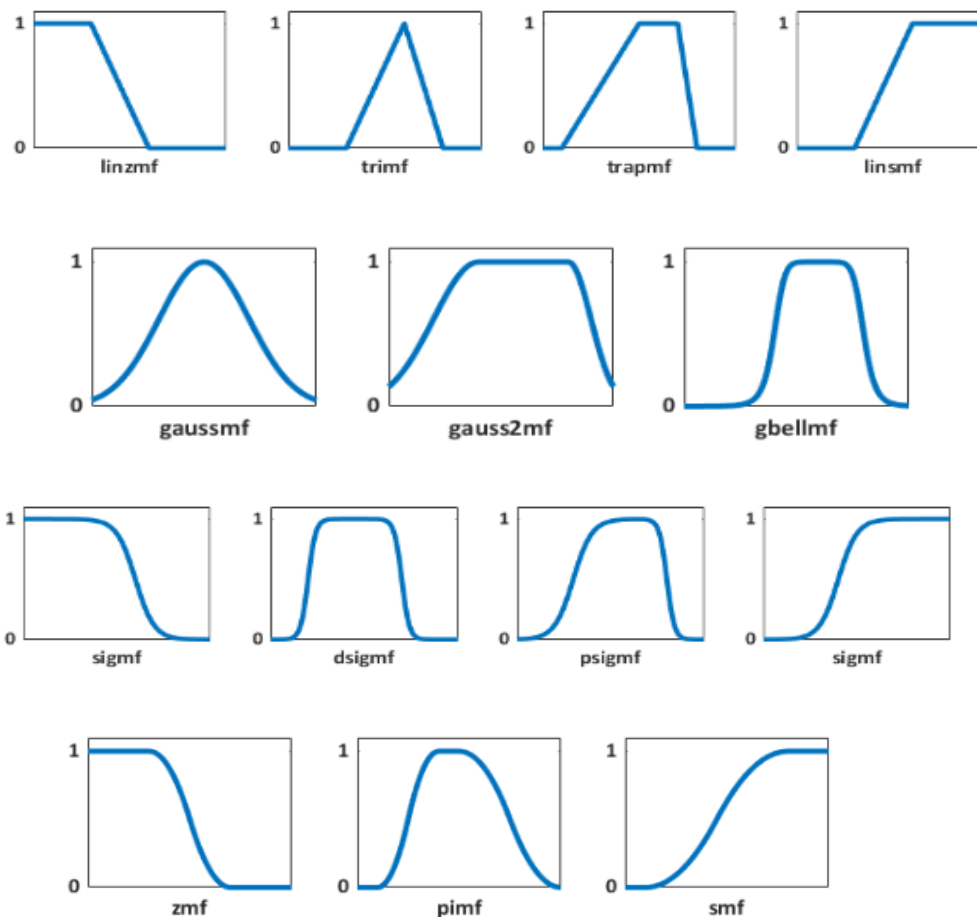
Obiective: utilizarea funcțiilor specifice pentru definirea mulțimilor fuzzy, înțelegerea operațiilor simple cu mulțimi fuzzy, familiarizarea cu modul de reprezentare a culorilor în MATLAB, înțelegerea operațiilor de bază pentru prelucrarea imaginilor și a conceptului de segmentare a unei imagini

Observație: MATLAB/Simulink se accesează online (<https://matlab.mathworks.com/>), prin logare cu credențialele MS Teams (cele de tip nume.prenume@student.utcluj.ro).

Termeni și acronime: MF, parametri ai unei mulțimi fuzzy, reuniune, intersecție, complement, segmentare

- Galeria de mulțimi fuzzy. Utilizarea și editarea unei mulțimi fuzzy.

MATLAB pune la dispoziție 13 tipuri de mulțimi fuzzy (MF – membership function):



Detalii despre orice mulțime fuzzy, împreună cu o mică secvență de cod, se accesează cu “*help nume_funcție*”, astfel:

```
help trapmf
```

Exercițiul 1

Rulați secvența următoare, ce are ca scop afișarea unei mulțimi fuzzy trapezoidale. Analizați fiecare linie de cod.

```
close all % inchide figura anterioara
clear all % sterge toate variabilele existente in workspace
clc      % sterge consola (fereastra de comenzi)
x = (0:0.1:10)'; % Universul discutiei este [0,10], in care se definesc
%puncte cu pasul de 0.1
params=[2 3 7 9];% parametrii multimii fuzzy trapezoidale
y = trapmf(x,params); % calculeaza valorile functiei de apartenenta
plot(x, y, 'linewidth',2);
axis([0 10 -0.1 1.1]);
xlabel('universul discutiei'); % numele variabilei pe axa orizontala
ylabel ('grad de apartenenta'); % numele variabilei pe axa verticala
set(gcf, 'name', 'Multime fuzzy trapezoidala', 'numbertitle', 'off'); %
%numele ferestrei
```

Exercițiul 2

Pe graficul afișat anterior, se adaugă reprezentarea gradului de apartenență al unui punct ales arbitrar la mulțimea fuzzy.

```
hold on
x1=2.75;
u1=evalmf(x1,params,'trapmf');
sprintf('Valoarea x1=%1.2f are gradul de apartenenta u1=%1.2f',x1,u1);
plot (x1,u1,'r*') % marcheaza punctul pe grafic
plot ([x1,x1],[0,u1],...
      'linestyle','-','color','r')
plot ([0,x1],[u1,u1],...
      'linestyle','-','color','r')
hold off
```

Exercițiul 3

Considerați variabila lingvistică “Viteza”, cu universul discuției [0, 140] km/h. Definiți cinci valori lingvistice și reprezentați grafic cele cinci mf corespunzătoare, pe aceeași axă (folosiți “*hold on*”). Mulțimile fuzzy trebuie să formeze o partiție fuzzy în intervalul [0,140]. Afișați la linia de comandă valorile gradelor de apartenență la fiecare mulțime a punctelor 10 km/h, 52 km/h, 85 km/h și 100 km/h.

○ Operații cu mulțimi fuzzy

Reuniunea a două mulțimi fuzzy A și B este definită cu ajutorul funcțiilor de apartenență:

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

Intersecția a două mulțimi fuzzy A și B este definită astfel:

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

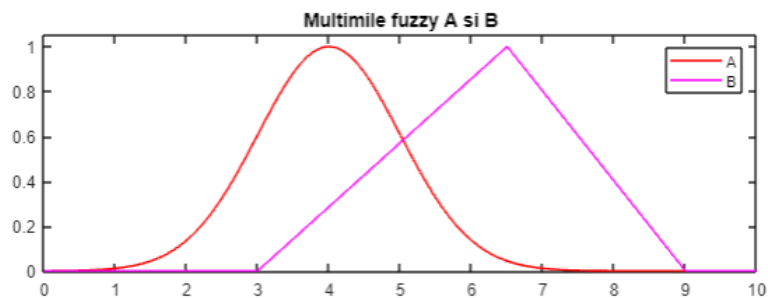
Complementul unei mulțimi fuzzy A este definit astfel:

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

Exercițiul 4

Rulați secvența următoare, pentru afișarea reuniunii a două mulțimi fuzzy. Analizați fiecare linie de cod.

```
close all
x = (0:0.1:10)'; % Universul discutiei este [0,10], in care se definesc
%puncte cu pasul de 0.1
u1=gaussmf(x,[1,4]); % prima mf, gaussiana
u2=trimf(x,[3 6.5 9]); % a doua mf, triunghiulara
u_reuniune=max(u1,u2); % calculeaza gradele de apartenenta ale reuniunii
%prin operatorul "MAX"
hold on
subplot(2,1,1); % imparte fereastra de reprezentare a unei figuri in doua
%regiunii= verticale
% graficul se afiseaza in fereastra de sus
plot(x,u1,'r');hold on
plot(x,u2,'m'); hold off
axis([0 10 0 1.05]);
legend('A','B');
title('Multimile fuzzy A si B')
subplot(2,1,2) % se va afisa in fereastra de jos
plot(x,u_reuniune,'color','b','linewidth',2)
axis([0 10 0 1.05]);
title('Rezultatul reuniunii')
set(gcf,'name','Reuniunea multimilor fuzzy A si B - operatorul "max"',
'numbertitle','off'); % numele ferestrei
```



Exercițiul 5

Modificați secvența de cod anterioară, astfel încât să afișați într-o singură fereastră, pe grafice diferite, în ordine: mulțimile fuzzy A și B, rezultatul reuniunii, rezultatul intersecției, rezultatul complementului.

- **Reprezentarea culorilor în MATLAB. Spațiile RGB și HSV.**

Spațiul **RGB** (red, green, blue) este definit de cele trei culori primare, roșu, verde și albastru, din care se pot obține, prin combinații aditive, toate celelalte culori ale spectrului. Valorile R, G și B sunt cuprinse în domeniul [0;1]. Pentru a reprezenta în MATLAB o culoare în spațiul RGB, se poate folosi numele (întreg sau prescurtat) al culorii, pentru culorile predefinite, sau un vector ce combină valorile componentelor primare.

Pentru a defini culoarea roz ca fond pentru fereastra curentă, scrieți în linia de comandă:

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

Exercițiul 6

Modificați culoarea de fond a ferestrei curente la cyan, respectiv verde, utilizând atât numele prescurtat al culorii, cât și reprezentarea vectorială.

În spațiul **HSV** (hue, saturation, value), culorile sunt reprezentate prin atributele perceptuale de nuanță (hue), saturație (saturation) și luminozitate sau valoare (value). Valorile H, S și V sunt reprezentate în domeniul [0;1]. În MATLAB, la deschiderea unei imagini color, reprezentarea sa implicită este în spațiul RGB. Conversia reprezentării în spațiul HSV se face cu funcția "rgb2hsv".

- **Transpunerea culorilor în mulțimi fuzzy**

Diferitele valori, între 0 și 1, ale nuanței, H, definesc culorile importante din spectrul vizibil: roșu, orange, galben, verde, cyan, albastru, indigo (purpuriu), violet (magenta), roz și (la capătul domeniului), din nou roșu. Pentru reprezentarea culorilor ca mulțimi fuzzy, universul discuției este domeniul valorilor posibile pentru nuanță, variabila din universul discuției fiind nuanță H. Ca domeniu de valori, se va utiliza același domeniu cu cel utilizat în prelucrarea digitală a imaginilor, intervalul $H=[0; 255]$ (reprezentare pe 8 biți). Pentru fiecare culoare, se definește câte o mulțime fuzzy peste X. Rezultă un total de 9 mulțimi fuzzy, cu valorile lingvistice: Red, Orange, Yellow, Green, Cyan, Blue, Purple, Magenta, Pink, definite peste H (intervalul [0; 255]), cu valori în intervalul [0; 1].

Exercițiul 7

Rulați secvența de mai jos pentru a vizualiza reprezentarea cu mulțimi fuzzy a culorilor. Analizați fiecare linie de cod.

```
close all;
clear all;
clc;
% definirea multimilor fuzzy pentru nuanta:
H=(0:.5:255)'; %universul discutiei
Red_l=trimf(H, [0 0 21]); % rosu, la stanga
Red_r=trimf(H, [234 255 255]); %rosu, la dreapta
```

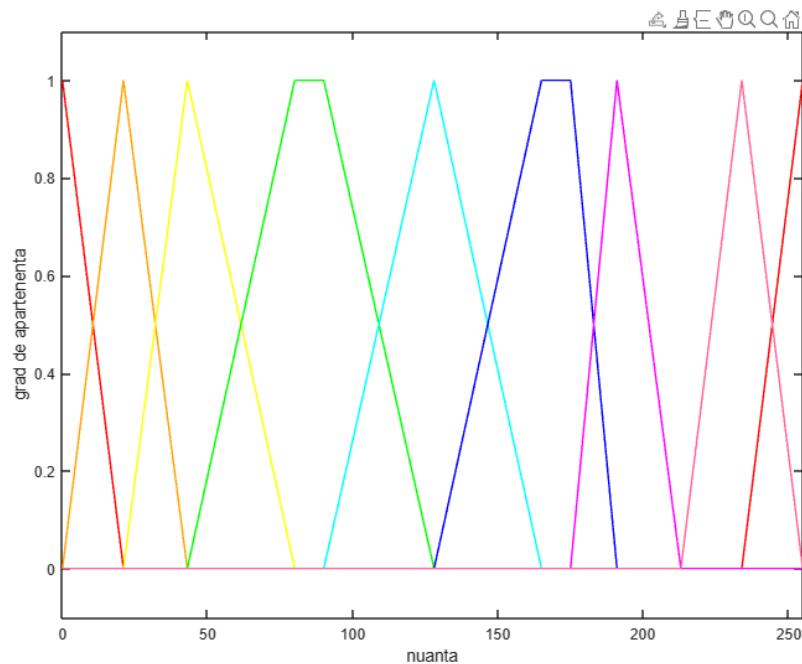
```

Red=max(Red_l, Red_r);
Orange=trimf(H, [0 21 43]); % portocaliu
Yellow=trimf(H, [21 43 80]); % galben
Green=trapmf(H, [43 80 90 128]); % verde
Cyan=trimf(H, [90 128 165]); % turcoaz
Blue=trapmf(H, [128 165 175 191]); % albastru
Purple=trimf(H, [175 191 213]); % violet
Pink=trimf(H, [213 234 255]); %roz

plot(H, Red_l, 'color', 'r');
axis ([0 255 -0.1 1.1]);
hold on;
xlabel('nuanta'); % numele variabilei pe axa orizontala
ylabel ('grad de apartenenta'); % numele variabilei pe axa verticala
set(gcf, 'name', 'Multimi fuzzy pentru reprezentarea culorilor',
'numbertitle', 'off'); % numele ferestrei

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

```



○ Segmentarea unei imagini color

Segmentarea unei imagini color presupune izolarea zonelor de o anumită culoare, față de celelalte zone. Pentru exemplificare, se va analiza o imagine din care se dorește separarea zonelor (obiectelor) de culoare dominant roșie, adică reprezentarea lor prin niveluri de gri cât mai apropiate de alb, spre deosebire de orice alte culori din imagine, care se vor reprezenta prin negru sau niveluri de gri cât mai apropiate de negru.

Pentru segmentare, se va forma mulțimea fuzzy *CloseToRed*, prin care se dorește capturarea zonelor din imagine care sunt nu doar roșu pur, ci și nuanțe apropiate de roșu. Mulțimea fuzzy *CloseToRed* este formată aplicând operațiile de bază (reuniune, intersecție, complement):

$$CloseToRed = Red \text{ SAU } (Orange \text{ SI NU } Yellow) \text{ SAU } Pink$$

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

Exercițiul 8

Rulați secvența de mai jos pentru afișarea mulțimii *CloseToRed*. Analizați fiecare linie de cod.

```
figure;
NOT_Yellow=1-Yellow;
OrangeNotYellow=min(Orange, NOT_Yellow);
CloseRed=max(Red, OrangeNotYellow);
CloseToRed=max(CloseRed, Pink);
plot(H, CloseToRed, 'Linewidth', 2, 'color', 'r');
axis ([0 255 -1 1.1]);
xlabel('nuanta'); % numele variabilei pe axa orizontala
ylabel ('grad de apartenenta'); % numele variabilei pe axa verticala
set(gcf, 'name', 'Multimi fuzzy CuloareAproapeRosu', 'numbertitle',
'off'); % numele ferestrei
```

Cu cât obiectul sau zona din imagine are nuanța mai apropiată de roșu, cu atât gradul său de apartenență la mulțimea *CloseToRed* este mai mare. Reprezentarea vizuală a rezultatului obținut se face prin scalarea gradului de apartenență rezultat în domeniul [0; 255], pentru fiecare pixel. Imaginea segmentată va conține zone albe sau aproape albe în locul obiectelor roșii sau aproape roșii.

Exercițiul 9

Descărcați arhiva "*imagini.zip*" și plasați conținutul acesteia (cu *drag-and-drop*) în directorul curent în care lucrează MATLAB. Dezarhivarea se face cu dublu click pe fișier.

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

Rulați secvența de mai jos pentru segmentarea unei imagini color. Analizați efectul segmentării asupra fiecărei imagini din arhivă.

```
close all; clc; clear all
% imagini disponibile; la un moment dat, se va selecta doar una dintre ele
img='Tiffany24.bmp';
% img='Cuticle.bmp';
% img='Lenna.bmp';
% img='Mouse.bmp';
% img='Peppers2.bmp';

OrigImg=imread(img); % citirea imaginii
```

```

HSVimg=(rgb2hsv(OrigImg)); % conversie RGB->HSV
Himg=(HSVimg(:,:,1)); % extragerea componenteii nuanta Hue, cuprinsa in
%intervalul [0,1]
H=(Himg*255); % normalizare la [0, 255]

% calculul gradului de apartenenta al nuantei pentru fiecare pixel, la
% fiecare multime
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]);

% normalizare la [0, 255]
RedObj=uint8(255*RedH);
% compunerea multimii "CloseToRed"
NOT_YellowH=1-YellowH;
OrangeNotYellowH=min(OrangeH, NOT_YellowH);
CloseRedH=max(RedH, OrangeNotYellowH);
CloseToRedH=max(CloseRedH, PinkH);
% afisare
imshow(OrigImg); set(gcf, 'name', 'Original image'); % imagine originala
figure
imshow(RedObj); set(gcf, 'name', 'Red segments'); % evidentiere obiecte
%rosii
figure
imshow(uint8(CloseToRedH*255)); set(gcf, 'name', 'Close to red
segments'); % evidentiere obiecte aproape rosii

```



Imagine originală



Imagine segmentată după roșu



Imagine segmentată după roșu și aproape roșu

Exercițiul 10

Creăți un script care să definească o mulțime fuzzy pentru aproximarea unei alte nuanțe. Afișați mulțimea fuzzy rezultată. Realizați segmentarea după nuanța și după aproximarea nuanței alese. Pentru testare, puteți utiliza imaginile din arhivă sau alte imagini reprezentative.