**App Designer - Matlab**

1. **Đọc ảnh, truy xuất thông tin ảnh**

f=app.UIFigure;

[filename,path]=uigetfile({'\*.png';'\*.jpg';'\*.tif'},'Chọn ảnh');

if isequal(filename,0)||isequal(path,0)

uialert(f,'Chưa chọn ảnh','Lỗi đọc ảnh')

cla(app.axe1);

else

file=strcat(path,filename);

[app.img1,app.map1]=imread(file);

imshow(app.img1,app.map1,'Parent',app.axe1);

info=imfinfo(file);

if length(info)>1

info=info(1);

end

app.colortype=info.ColorType;

app.bitdepth=info.BitDepth;

end

**Property % Description**

img1

map1

colortype

bitdepth

gray

bin

**Chuyển đổi ảnh (RGB/indexed => grayscale; grayscale => binary)**

**\*Đa cấp xám**

if app.colortype=="indexed"

app.I=ind2gray(app.img1,app.map1);

imshow(app.I,'Parent',app.axe2);

elseif strcmp(info.ColorType,'truecolor')

app.I=rgb2gray(app.img1);

imshow(app.I,'Parent',app.axe2);

else

app.I=app.img1;

imshow(app.I,'Parent',app.axe2);

end

**\*Nhị phân**

app.B=imbinarize(app.I);

imshow(app.B,'Parent',app.axe3);

1. **Xử lý điểm:**
   * **Ảnh âm bản** *(có thể yêu cầu tự viết hàm hoặc dùng hàm dựng sẵn)*

iab=imcomplement(app.img1);

imshow(iab,'Parent',app.axe4);

* + **Phân ngưỡng ảnh toàn cục (ngưỡng đơn hoặc ngưỡng kép)** *(yêu cầu tự viết hàm)*

1. **Xử lý tổ chức đồ:**
   * **Hiển thị tổ chức đồ của ảnh**

[pix,gl]=imhist(app.img1);

bar(app.axe5,gl,pix,0.2,"stacked","black");

**% Hist RGB**

red\_hist = imhist(red(:,:,1));

green\_hist = imhist(green(:,:,2));

blue\_hist = imhist(blue(:,:,3));

**% Hist CMY**

cyan\_hist = imhist(CMY(:,:,1));

magenta\_hist = imhist(CMY(:,:,2));

yello\_hist = imhist(CMY(:,:,3));

* + **Trượt, căng, biến đổi tuyến tính, cân bằng tổ chức đồ** *(có thể yêu cầu tự viết hàm hoặc dùng hàm dựng sẵn)*

**Hàm trượt histogram**

function [oimg,omap] = truot\_histogram (file, c)

[img,map]=imread(file);

info=imfinfo(file);

if length (info)>1

info=info(1);

end

lmax=2^info.BitDepth-1;

[y1,x1]=size(img);

oimg=zeros(y1,x1);

omap=map;

for i=1:y1

for j=1:x1

oimg(i,j)= img(i,j)+c;

if oimg(i,j)>lmax

oimg(i,j)=lmax;

end

if oimg(i,j)<0

oimg(i,j)=0;

end

end

end

end

**gọi hàm**

if app.drd\_xld.Value=="Truot to chuc do"

c=app.Slider.Value;

[im,m] = truot\_histogram(app.file,c);

imshow(im,m,'parent',app.axe\_kq\_xld);

**- Căng tổ chức đồ**:

**Hàm căng histogram**

function [oimg,omap] = cang\_histogram (file, c)

[img,map]=imread(file);

info=imfinfo(file);

if length (info)>1

info=info(1);

end

lmax=2^info.BitDepth-1;

[y1,x1]=size(img);

oimg=zeros(y1,x1);

omap=map;

for i=1:y1

for j=1:x1

oimg(i,j)= round(img(i,j)\*c);

if oimg(i,j)>lmax

oimg(i,j)=lmax;

end

end

end

end

**gọi hàm:**

if app.drd\_xld.Value=="Cang to chuc do"

c=app.txt\_c.Value;

[im,m]=cang\_histogram(app.file,c);

imshow(im,m,'parent',app.axe\_kq\_xld);

1. **Các phép toán số học trên ảnh** *(có thể yêu cầu tự viết hàm hoặc dùng hàm dựng sẵn)*
   * **Chuyển ảnh về cùng kích thước** *(chuyển thủ công hoặc dùng hàm imresize với kích thước ymax, xmax)*

inSize = size(app.img1);

app.outsizeIm1 = imresize(app.I,inSize);

app.outsizeIm2 = imresize(app.I2,inSize);

imshow(app.outsizeIm1,app.map1,'Parent',app.axe3);

imshow(app.outsizeIm2,app.map2,'Parent',app.axe4);

* + **Cộng ảnh**

cong=imadd(app.outsizeIm1,app.outsizeIm2);

imshow(cong ,'Parent',app.axe5);

* + **Trừ ảnh**

tru=imsubtract(app.outsizeIm1,app.outsizeIm2);

imshow(tru ,'Parent',app.axe6);

1. **Lọc làm mịn ảnh: mean, gaussian, median, order filter**
   * Mean, Gaussian: xây dựng mặt nạ lọc (dùng ma trận hoặc hàm fspecial), sử dụng các hàm hỗ trợ lọc ảnh dựng sẵn (imfilter) với các mặt nạ lọc đã xây dựng

**Mean**

app.I=app.img1;

fs=app.Spinner.Value;

h= fspecial('average',fs);

out=imfilter(app.I,h);

imshow(out,'Parent',app.axe2);

**Gaussian**

app.I=app.img1;

sigma=app.Spinner.Value;

out=imgaussfilt(app.I,sigma);

imshow(out,'Parent',app.axe3);

* + Median và order filter: sử dụng các hàm dựng sẵn hoặc tự xây dựng hàm.

**Median**

app.I=app.img1;

if app.DropDown.Value=="3x3"

n=3;

elseif app.DropDown.Value=="5x5"

n=5;

elseif app.DropDown.Value=="7x7"

n=7;

else

n=9;

end

out=medfilt2(app.I,[n,n]);

imshow(out,'Parent',app.axe4);

**order filter**

app.I=app.img1;

fs=app.Spinner.Value;

order=floor(fs\*fs/2);

out=ordfilt2(app.I,order,ones(fs));

imshow(out,'Parent',app.axe5);

1. **Phát hiện biên + lọc làm nét ảnh: Robert, Prewitt, Sobel, Laplacian**
   * **Robert, Prewitt, Sobel: tạo cặp mặt nạ lọc Gx và Gy; lọc ảnh với Gx và Gy**

**\*Chuyển đa cấp xám**

**Tách biên Gx,Gy + Toàn biên (app.I2: chổ lưu ảnh đa cấp xám)**

img=app.I;

if app.ddtt2.Value=="Robert"

hx=[0 1; -1 0];

f=imfilter(img,hx);%filter2(hx,img);

imshow(f,'Parent',app.axe23);

hy=[1 0; 0 -1];

f2=imfilter(img,hy);%filter2(hy,img);

imshow(f2,'Parent',app.axe24);

app.f3=imadd(f,f2);

imshow(app.f3,'Parent',app.axe25);

elseif app.ddtt2.Value=="Prewitt"

hx=[-1 0 1; -1 0 1; -1 0 1];

f=imfilter(img,hx);%filter2(hx,img);

imshow(f,'Parent',app.axe23);

hy=[-1 -1 -1; 0 0 0; 1 1 1];

f2=imfilter(img,hy);%filter2(hy,img);

imshow(f2,'Parent',app.axe24);

app.f3=imadd(f,f2);

imshow(app.f3,'Parent',app.axe25);

else

hx=[-1 0 1; -2 0 2; -1 0 1];

f=imfilter(img,hx);%filter2(hx,img);

imshow(f,'Parent',app.axe23);

hy=[-1 -2 -1; 0 0 0; 1 2 1];

f2=imfilter(img,hy);%filter2(hy,img);

imshow(f2,'Parent',app.axe24);

app.f3=imadd(f,f2);

imshow(app.f3,'Parent',app.axe25);

end

**Làm nét**

img=app.I2;

f=imadd(img,app.f3);

imshow(f,'Parent',app.axe26)

* + **Tùy biến bộ lọc Laplacian với các mặt nạ mở rộng**

**Toàn biên**

img=app.I;

if app.ddmnl.Value=="-4"

h=[0 1 0; 1 -4 -1; 0 1 0];

elseif app.ddmnl.Value=="-8"

h=[1 1 1; 1 -8 1; 1 1 1];

elseif app.ddmnl.Value=="4"

h=[0 -1 0; -1 4 -1; 0 -1 0];

else

h=[-1 -1 -1; -1 8 -1; -1 -1 -1];

end

app.f1=imfilter(img,h);

app.f1=uint8(app.f1);

imshow(app.f1,'Parent',app.axe3);

**Làm nét**

img=app.I;

if app.ddmnl.Value=="4" || app.ddmnl.Value=="8"

f4=imadd(img,app.f1);

else

f4=imsubtract(img,app.f1);

end

imshow(f4,'Parent',app.axe4);

1. **Xử lý ảnh trên miền tần số**
   * Xây dựng bộ lọc thông thấp Ideal, Gaussian, Butterworth

**Property % Description**

img1

map1

colortype

bitdepth

gray

bin

I

FT

FTzero

H

rs

iFTzero

iFT

**Đọc ảnh**

f=app.UIFigure;

[filename,path]=uigetfile({'\*.png';'\*.jpg';'\*.tif'},'Chọn ảnh');

if isequal(filename,0)||isequal(path,0)

uialert(f,'Chưa chọn ảnh','Lỗi đọc ảnh')

cla(app.axe1);

else

file=strcat(path,filename);

[app.img1,app.map1]=imread(file);

%imshow(app.img1,app.map1,'Parent',app.axe1);

info=imfinfo(file);

if length(info)>1

info=info(1);

end

app.colortype=info.ColorType;

app.bitdepth=info.BitDepth;

end

if app.colortype=="indexed"

%[X,map]=imread(file);

app.I=ind2gray(app.img1,app.map1);

imshow(app.I,'Parent',app.axe1);

elseif strcmp(info.ColorType,'truecolor')

app.I=rgb2gray(app.img1);

imshow(app.I,'Parent',app.axe1);

else

app.I=app.img1;

imshow(app.I,'Parent',app.axe1);

end

X=double(app.I);

app.FT=fft2(X);

imshow(app.FT,'Parent',app.axe2); **%Ảnh tần số**

ampi=log(1+abs(app.FT));

minvalue=min(min(ampi));

maxvalue=max(max(ampi));

imshow(ampi,[minvalue maxvalue],'Parent',app.axe3);**%Ảnh phổ tăng cường**

app.FTzero=fftshift(app.FT);

ampi=log(1+abs(app.FTzero));

minvalue=min(min(ampi));

maxvalue=max(max(ampi));

imshow(ampi,[minvalue maxvalue],'Parent',app.axe4); **%Ảnh hướng tâm**

**Tạo mặt nạ**

D0 = app.txtbk.Value;

[M,N]=size(app.FTzero);

app.H=zeros(M,N);

if app.ddboloc.Value=="Ideal"

for u=1:M

for v=1:N

**if sqrt((u-M/2)^2+(v-N/2)^2)<=D0**

**app.H(u,v)=1;**

end

end

end

end

if app.ddboloc.Value=="Gaussian"

mau = 2\*D0\*D0;

for u = 1:M

for v = 1:N

**tu = sqrt((u - M/2)^2 + (v - N/2)^2);**

**app.H(u,v)= exp(-(tu^2)/mau);**

end

end

end

if app.ddboloc.Value=="Butterworth"

n = app.txtn.Value;

for u = 1:M

for v = 1:N

**D = sqrt((u - M/2)^2 + (v - N/2)^2);**

**app.H(u,v) = 1 / (1 + (D / D0)^(2\*n));**

end

end

end

imshow(app.H,'Parent',app.axe5) **%Bộ lọc**

app.rs=app.FTzero.\*app.H;

ampi=log(1+abs(app.rs));

minvalue=min(min(ampi));

maxvalue=max(max(ampi));

imshow(ampi,[minvalue maxvalue],'Parent',app.axe6); **%Ảnh lọc miền tần số**

app.iFTzero=ifftshift(app.rs);

ampi=log(1+abs(app.iFTzero));

minvalue=min(min(ampi));

maxvalue=max(max(ampi));

imshow(ampi,[minvalue maxvalue],'Parent',app.axe7); **%Ảnh phổ về gốc**

**Làm mịn**

app.iFT=ifft2(app.iFTzero);

imshow(app.iFT,[],'Parent',app.axe8);

* + Xây dựng bộ lọc thông cao Ideal, Gaussian, Butterworth

**Tạo mặt nạ**

Ideal:

if(u-M/2)^2 +(v-N/2)^2 <= D0**^2**

app.boloc\_tc(u,v)=0;

else

app.boloc\_tc(u,v)=1;

end

Gaussian:

tu = sqrt((u - M/2)^2 + (v - N/2)^2);

app.boloc\_tc(u,v)= 1-exp(-(tu^2)/mau);

Butterworth:

D = sqrt((u - M/2)^2 + (v - N/2)^2);

app.boloc\_tc(u,v) = 1 / (1 + (D0 / D)^(2\*n));

1. **Xử lý hình thái (đọc ảnh đầu vào 🡪 ảnh nhị phân)**

if app.colortype=="indexed"

app.I=ind2gray(app.img1,app.map1);

app.B=imbinarize(app.I);

elseif strcmp(info.ColorType,'truecolor')

app.I=rgb2gray(app.img1);

app.B=imbinarize(app.I);

else

app.I=app.img1;

app.B=app.I;

end

imshow(app.B,'Parent',app.axe1);

* + Ăn mòn, làm dày ảnh

**Làm dày**

day=app.B;

sq=ones(3,3);

td=imdilate(day,sq);

imshow(td,'Parent',app.axe2);

**Ăn mòn**

mon=app.B;

sq=ones(3,3);

ce=imerode(mon,sq);

imshow(ce,'Parent',app.axe3);

* + Đóng ảnh, mở ảnh

**Đóng ảnh**

mon=app.B;

sq=ones(3,3);

ce=imerode(mon,sq);

td=imdilate(ce,sq);

imshow(td,'Parent',app.axe4);

**Mở ảnh**

day=app.B;

sq=ones(3,3);

td=imdilate(day,sq);

ce=imerode(td,sq);

imshow(ce,'Parent',app.axe5);

1. **Thao tác với ảnh màu RGB**
   * Tách và hiển thị ảnh trên 3 kênh màu R, G, B hoặc trên 3 kênh màu bù C, M, Y

**Tách ảnh RGB**

if app.colortype ~= "truecolor"

uialert(f,'phai chon anh RGB','loi chon anh')

cla(app.axegoc)

else

imshow(app.img, app.map, 'Parent', app.axegoc);

end

app.red=app.img1;

app.red(:,:,2)=0;

app.red(:,:,3)=0;

app.green=app.img1;

app.green(:,:,1)=0;

app.green(:,:,3)=0;

app.blue=app.img1;

app.blue(:,:,1)=0;

app.blue(:,:,2)=0;

imshow(app.red,'Parent',app.axe1\_4);

imshow(app.green,'Parent',app.axe1\_5);

imshow(app.blue,'Parent',app.axe1\_6);

**Tách ảnh CMY**

cyan=app.img;

cyan(:,:,1)=0;

magenta=app.img;

magenta(:,:,2)=0;

yellow= app.img;

yellow(:,:,3)=0;

imshow(cyan,'Parent',app.axeblue);

imshow(magenta,'Parent',app.axered);

imshow(yellow,'Parent',app.axegreen);

* + **Tạo lại ảnh RGB**

r=app.red(:,:,1);

g=app.green(:,:,2);

b=app.blue(:,:,3);

out=cat(3,r,g,b);

imshow(out,'Parent',app.axe1\_2);

* + **Tạo ảnh CMY**

CMY = app.RGB;

CMY(:,:,1) = 255 - app.RGB(:,:,1);

CMY(:,:,2) = 255 - app.RGB(:,:,2);

CMY(:,:,3) = 255 - app.RGB(:,:,3);

imshow(CMY,'Parent',app.axecmy);

1. Sử dụng các components trong Component Library của MATLAB App Designer để tuỳ biến các thông số, bộ lọc; thiết kế giao diện đảm bảo tính tương tác tốt với người dùng.