# 第一题

## 代码

clear

close all

clc

%% 灰度化并二值化

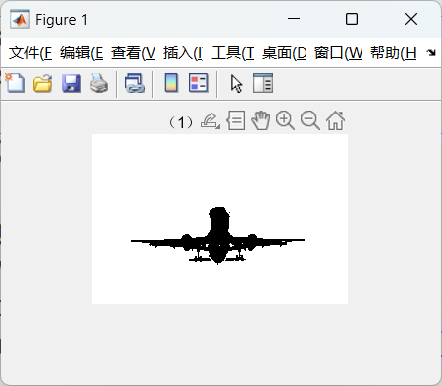
image=imread("plane.JPG");

gray=rgb2gray(image);

bw=imbinarize(gray);

figure,imshow(bw),title("（1）二值化图像")

## 运行结果



# 第二题

## 代码

clear

close all

clc

%% 形态学滤波

image=imread("plane.JPG");

gray=rgb2gray(image);

bw=imbinarize(gray);

se=strel('square',3);

morph=imopen(bw,se);

morph=imclose(morph,se);

figure,imshow(morph),title('（2）形态学滤波');

## 运行结果

图形用户界面, 应用程序

描述已自动生成

# 第三题

## 代码

clear

close all

clc

%% 边缘跟踪，用红色线标注

image=imread("plane.JPG");

gray=rgb2gray(image);

bw=imbinarize(gray);

se=strel('square',3);

morph=imopen(bw,se);

morph=imclose(morph,se);

[boundaries,labels]=bwboundaries(1-morph);

figure,imshow(labels),title('（3）划分的区域');

hold on;

for i=1:length(boundaries)

boundary=boundaries{i};

plot(boundary(:,2),boundary(:,1),'r','LineWidth',2);

end

## 运行结果

图形用户界面, 应用程序

描述已自动生成

# 第四题

## 代码

clear

close all

clc

%% 计算各区域边界点的傅里叶描绘子并用四分之一点重建边界

image=imread("plane.JPG");

gray=rgb2gray(image);

bw=imbinarize(gray);

se=strel('square',3);

morph=imopen(bw,se);

morph=imclose(morph,se);

boundaries=bwboundaries(1-morph);

M=zeros(length(boundaries));

for k=1:length(boundaries)

N=length(boundaries{k});

if N/2~=round(N/2)

boundaries{k}(end+1,:)=boundaries{k}(end,:);

N=N+1;

end

M(k)=N\*3/4;

end

background=zeros(size(morph));

figure,imshow(background),title("（4）重建边界");

hold on;

for k=1:length(boundaries)

z=boundaries{k}(:,2)+1i\*boundaries{k}(:,1);

Z=fft(z);

[Y,I]=sort(abs(Z));

for count=1:M(k)

Z(I(count))=0;

end

zz=ifft(Z);

plot(real(zz),imag(zz),'w');

end

## 运行结果

图形用户界面, 应用程序, PowerPoint

描述已自动生成

# 第五题

## 代码

clear

close all

clc

%% 拓展内容

%% 二值化

image=imread("plane.JPG");

gray=rgb2gray(image);

bw=imbinarize(gray,"adaptive");

figure,imshow(bw),title("（5）adaptive 方法二值化图像")

%% 形态学滤波

bw=imbinarize(gray,"global");

se=strel('arbitrary',3);

morph=imopen(bw,se);

morph=imclose(morph,se);

figure,imshow(morph),title('（6）arbitrary 形态学滤波');

%% 重建边界

boundaries=bwboundaries(1-morph);

M=zeros(length(boundaries), 4);

for k=1:length(boundaries)

N=length(boundaries{k});

if N/2~=round(N/2)

boundaries{k}(end+1,:)=boundaries{k}(end,:);

N=N+1;

end

M(k,:)=[N/2 N\*7/8 N\*15/16 N\*63/64]; % N/2,N/8,N/16,N/64 项重建

end

background=zeros(size(morph));

count=7;

points=["N/2","N/8","N/16","N/64"];

for m = 1:4

count=count+1;

figure,imshow(background),title(strcat("（",string(count),"）",points{m}," 项重建"));

hold on;

for k=1:length(boundaries)

z=boundaries{k}(:,2)+1i\*boundaries{k}(:,1);

Z=fft(z);

[Y,I]=sort(abs(Z));

for count=1:M(k,m)

Z(I(count))=0;

end

zz=ifft(Z);

plot(real(zz),imag(zz),'w');

end

end

%% 图像分割

C = makecform('srgb2lab'); %设置转换格式

imageLab = applycform(image, C);

ab = double(imageLab(:,:,2:3)); %取出 lab 空间的 a 分量和 b 分量

nrows = size(ab,1);

ncols = size(ab,2);

ab = reshape(ab,nrows\*ncols,2);

nColors = 3; % 分割的区域个数为 3

[clusterIdx,clusterCenter] = kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3); % 重复聚类 3 次

pixelLabels = reshape(clusterIdx,nrows,ncols);

% 显示分割后的各个区域

segmentedImages = cell(1,3);

rgbLabel = repmat(pixelLabels,[1 1 3]);

for k = 1:nColors

color = image;

color(rgbLabel ~= k) = 0;

segmentedImages{k} = color;

end

figure;

subplot(1,3,1),imshow(segmentedImages{1}), title("（10）分割区域 1");

subplot(1,3,2),imshow(segmentedImages{2}), title("（11）分割区域 2");

subplot(1,3,3),imshow(segmentedImages{3}), title("（12）分割区域 3");

## 运行结果

图形用户界面, 应用程序

描述已自动生成 图形用户界面, 应用程序, PowerPoint

描述已自动生成 图形用户界面, 应用程序

描述已自动生成