|  |
| --- |
| **实验摘要：**  利用MATLAB完成图形的处理，傅里叶变换和反变换，高低频提取与合并。  频域制作数字盲水印和去除数字盲水印 |
| **实验题目**  1. 图片的高频信息与低频信息  合成图片。找两张轮廓比较像的图片A和B，有一张是你本人。提取一张照片的低频信息，另一张图片的高频信息，结合这两个照片。设置不同的频率门限，组合照片，组合的效果是，放大看是A，缩小看是B。例如以下两张图片    分解图片。把下图爱因斯坦和玛丽莲梦露分开（此图缩小是玛丽莲梦露，截取不同的频段）  （MATLAB和Python均可实现）  2. 频域制作数字盲水印和去除数字盲水印 https://www.zhihu.com/question/50735753，看懂，想想，有想法写出来，做一个好玩的东西  3.    4. |
| **实验内容：**  一、  结果：    分解：  低频：    低频：  代码：  图片合成：  close all;  clear all;  img1=imread('1.jpg');  img2=imread('2.jpg');  img1=rgb2gray(img1);  img2=rgb2gray(img2);  g1=fft2(double(img1));  g2=fft2(double(img2));  g1=fftshift(g1);  g2=fftshift(g2);  [N1,N2]=size(g1);  n=2;  d0=20;  n1=fix(N1/2);  n2=fix(N2/2);  for i=1:N1  for j=1:N2  d=sqrt((i-n1)^2+(j-n2)^2);  h=exp(-d\*d/(2\*d0\*d0));  result1(i,j)=h\*g1(i,j);  end  end  result1=ifftshift(result1);  [N1,N2]=size(g2);  n=2;  d0=20;  n1=fix(N1/2);  n2=fix(N2/2);  for i=1:N1  for j=1:N2  d=sqrt((i-n1)^2+(j-n2)^2);  h=1-exp(-d\*d/(2\*d0\*d0));  result2(i,j)=h\*g2(i,j);  end  end  result2=ifftshift(result2);  X2=ifft2(result1+result2);  final=uint8(real(X2));  figure,imshow(final);  图片分解：  高频：  close all;  clear all;  img=imread('final.jpg');  img=rgb2gray(img);  figure,imshow(img);  g=fft2(double(img));  g=fftshift(g);  imshow(abs(g),[])  [N1,N2]=size(g);  n=2;  d0=20;  n1=fix(N1/2);  n2=fix(N2/2);  for i=1:N1  for j=1:N2  d=sqrt((i-n1)^2+(j-n2)^2);  h=1-exp(-d\*d/(2\*d0\*d0));  result(i,j)=h\*g(i,j);  end  end  result=ifftshift(result);  X2=ifft2(result);  final=uint8(real(X2));  figure,imshow(final);  低频：close all;  clear all;  img=imread('final.jpg');  img=rgb2gray(img);  figure,imshow(img);  g=fft2(double(img));  g=fftshift(g);  imshow(abs(g),[])  [N1,N2]=size(g);  n=2;  d0=20;  n1=fix(N1/2);  n2=fix(N2/2);  for i=1:N1  for j=1:N2  d=sqrt((i-n1)^2+(j-n2)^2);  h=exp(-d\*d/(2\*d0\*d0));  result(i,j)=h\*g(i,j);  end  end  result=ifftshift(result);  X2=ifft2(result);  final=uint8(real(X2));  figure,imshow(final);  二、  图片：  原图：    水印：    合成：    分解出水印：    代码：  clc;clear;close all;  alpha = 1;  im = double(imread('original image.jpg'))/255;  mark = double(imread('watermark.jpg'))/255;  figure, imshow(im),title('original image');  figure, imshow(mark),title('watermark');    imsize = size(im);  %random  TH=zeros(imsize(1),imsize(2),imsize(3));  TH1 = TH;  TH1(1:size(mark,1),1:size(mark,2),:) = mark;  M=randperm(imsize(1));  N=randperm(imsize(2));  save('encode.mat','M','N');  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  TH(i,j,:)=TH1(M(i),N(j),:);  end  end  % symmetric  mark\_ = zeros(imsize(1),imsize(2),imsize(3));  mark\_(1:imsize(1),1:imsize(2),:)=TH;  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  mark\_(imsize(1)+1-i,imsize(2)+1-j,:)=TH(i,j,:);  end  end  figure,imshow(mark\_),title('encoded watermark');  %imwrite(mark\_,'encoded watermark.jpg');    %% add watermark  FA=fft2(im);  figure,imshow(FA);title('spectrum of original image');  FB=FA+alpha\*double(mark\_);  figure,imshow(FB); title('spectrum of watermarked image');  FAO=ifft2(FB);  figure,imshow(FAO); title('watermarked image');  %imwrite(uint8(FAO),'watermarked image.jpg');  RI = FAO-double(im);  figure,imshow(uint8(RI)); title('residual');  %imwrite(uint8(RI),'residual.jpg');  xl = 1:imsize(2);  yl = 1:imsize(1);  [xx,yy] = meshgrid(xl,yl);  figure, plot3(xx,yy,FA(:,:,1).^2+FA(:,:,2).^2+FA(:,:,3).^2),title('spectrum of original image');  figure, plot3(xx,yy,FB(:,:,1).^2+FB(:,:,2).^2+FB(:,:,3).^2),title('spectrum of watermarked image');  figure, plot3(xx,yy,FB(:,:,1).^2+FB(:,:,2).^2+FB(:,:,3).^2-FA(:,:,1).^2+FA(:,:,2).^2+FA(:,:,3).^2),title('spectrum of watermark');    %% extract watermark  FA2=fft2(FAO);  G=(FA2-FA)/alpha;  GG=G;  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  GG(M(i),N(j),:)=G(i,j,:);  end  end  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  GG(imsize(1)+1-i,imsize(2)+1-j,:)=GG(i,j,:);  end  end  figure,imshow(GG);title('extracted watermark');  %imwrite(uint8(GG),'extracted watermark.jpg');    %% MSE and PSNR  C=double(im);  RC=double(FAO);  MSE=0; PSNR=0;  for i=1:imsize(1)  for j=1:imsize(2)  MSE=MSE+(C(i,j)-RC(i,j)).^2;  end  end  MSE=MSE/360.^2;  PSNR=20\*log10(255/sqrt(MSE));  MSE  PSNR    %% attack test  %% attack by smearing  %A = double(imread('gl1.jpg'));  %B = double(imread('attacked image.jpg'));  attack = 1-double(imread('attack.jpg'))/255;  figure,imshow(attack);  FAO\_ = FAO;  for i=1:imsize(1)  for j=1:imsize(2)  if attack(i,j,1)+attack(i,j,2)+attack(i,j,3)>0.5  FAO\_(i,j,:) = attack(i,j,:);  end  end  end  figure,imshow(FAO\_);  %extract watermark  FA2=fft2(FAO\_);  G=(FA2-FA)\*2;  GG=G;  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  GG(M(i),N(j),:)=G(i,j,:);  end  end  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  GG(imsize(1)+1-i,imsize(2)+1-j,:)=GG(i,j,:);  end  end  figure,imshow(GG);title('extracted watermark');    %% attack by cutting  s2 = 0.8;  FAO\_ = FAO;  FAO\_(:,s2\*imsize(2)+1:imsize(2),:) = FAO\_(:,1:int32((1-s2)\*imsize(2)),:);  figure,imshow(FAO\_);  %extract watermark  FA2=fft2(FAO\_);  G=(FA2-FA)\*2;  GG=G;  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  GG(M(i),N(j),:)=G(i,j,:);  end  end  for i=1:imsize(1)\*0.5  for j=1:imsize(2)  GG(imsize(1)+1-i,imsize(2)+1-j,:)=GG(i,j,:);  end  end  figure,imshow(GG);title('extracted watermark');      %%Ð¡²¨±ä»»¼ÓË®Ó¡£¬½âË®Ó¡´ó¼Ò°´ÕÕ¼ÓµÄË¼Â·Äæ¹ýÀ´¾ÍºÃ  clc;clear;close all;  %% read data  im = double(imread('gl1.jpg'))/255;  mark = double(imread('watermark.jpg'))/255;  figure, imshow(im),title('original image');  figure, imshow(mark),title('watermark');  %% RGB division  im=double(im);  mark=double(mark);  imr=im(:,:,1);  markr=mark(:,:,1);  img=im(:,:,2);  markg=mark(:,:,2);  imb=im(:,:,3);  markb=mark(:,:,3);  %% parameter  r=0.04;  g = 0.04;  b = 0.04;  %% wavelet tranform and add watermark  % for red  [Cwr,Swr]=wavedec2(markr,1,'haar');  [Cr,Sr]=wavedec2(imr,2,'haar');  % add watermark  Cr(1:size(Cwr,2)/16)=...  Cr(1:size(Cwr,2)/16)+r\*Cwr(1:size(Cwr,2)/16);  k=0;  while k<=size(Cr,2)/size(Cwr,2)-1  Cr(1+size(Cr,2)/4+k\*size(Cwr,2)/4:size(Cr,2)/4+...  (k+1)\*size(Cwr,2)/4)=Cr(1+size(Cr,2)/4+...  k\*size(Cwr,2)/4:size(Cr,2)/4+(k+1)\*size(Cwr,2)/4)+...  r\*Cwr(1+size(Cwr,2)/4:size(Cwr,2)/2);  Cr(1+size(Cr,2)/2+k\*size(Cwr,2)/4:size(Cr,2)/2+...  (k+1)\*size(Cwr,2)/4)=Cr(1+size(Cr,2)/2+...  k\*size(Cwr,2)/4:size(Cr,2)/2+(k+1)\*size(Cwr,2)/4)+...  r\*Cwr(1+size(Cwr,2)/2:3\*size(Cwr,2)/4);  Cr(1+3\*size(Cwr,2)/4+k\*size(Cwr,2)/4:3\*size(Cwr,2)/4+...  (k+1)\*size(Cwr,2)/4)=Cr(1+3\*size(Cr,2)/4+...  k\*size(Cwr,2)/4:3\*size(Cr,2)/4+(k+1)\*size(Cwr,2)/4)+...  r\*Cwr(1+3\*size(Cwr,2)/4:size(Cwr,2));  k=k+1;  end;  Cr(1:size(Cwr,2)/4)=Cr(1:size(Cwr,2)/4)+r\*Cwr(1:size(Cwr,2)/4);    % for green  [Cwg,Swg]=WAVEDEC2(markg,1,'haar');  [Cg,Sg]=WAVEDEC2(img,2,'haar');  Cg(1:size(Cwg,2)/16)=...  Cg(1:size(Cwg,2)/16)+g\*Cwg(1:size(Cwg,2)/16);  k=0;  while k<=size(Cg,2)/size(Cwg,2)-1  Cg(1+size(Cg,2)/4+k\*size(Cwg,2)/4:size(Cg,2)/4+...  (k+1)\*size(Cwg,2)/4)=Cg(1+size(Cg,2)/4+...  k\*size(Cwg,2)/4:size(Cg,2)/4+(k+1)\*size(Cwg,2)/4)+...  g\*Cwg(1+size(Cwg,2)/4:size(Cwg,2)/2);  Cg(1+size(Cg,2)/2+k\*size(Cwg,2)/4:size(Cg,2)/2+...  (k+1)\*size(Cwg,2)/4)=Cg(1+size(Cg,2)/2+...  k\*size(Cwg,2)/4:size(Cg,2)/2+(k+1)\*size(Cwg,2)/4)+...  g\*Cwg(1+size(Cwg,2)/2:3\*size(Cwg,2)/4);  Cg(1+3\*size(Cg,2)/4+k\*size(Cwg,2)/4:3\*size(Cg,2)/4+...  (k+1)\*size(Cwg,2)/4)=Cg(1+3\*size(Cg,2)/4+...  k\*size(Cwg,2)/4:3\*size(Cg,2)/4+(k+1)\*size(Cwg,2)/4)+...  g\*Cwg(1+3\*size(Cwg,2)/4:size(Cwg,2));  k=k+1;  end;  Cg(1:size(Cwg,2)/4)=Cg(1:size(Cwg,2)/4)+g\*Cwg(1:size(Cwg,2)/4);    % for blue  [Cwb,Swb]=WAVEDEC2(markb,1,'haar');  [Cb,Sb]=WAVEDEC2(imb,2,'haar');  Cb(1:size(Cwb,2)/16)+b\*Cwb(1:size(Cwb,2)/16);  k=0;  while k<=size(Cb,2)/size(Cwb,2)-1  Cb(1+size(Cb,2)/4+k\*size(Cwb,2)/4:size(Cb,2)/4+...  (k+1)\*size(Cwb,2)/4)=Cb(1+size(Cb,2)/4+...  k\*size(Cwb,2)/4:size(Cb,2)/4+(k+1)\*size(Cwb,2)/4)+...  g\*Cwb(1+size(Cwb,2)/4:size(Cwb,2)/2);  Cb(1+size(Cb,2)/2+k\*size(Cwb,2)/4:size(Cb,2)/2+...  (k+1)\*size(Cwb,2)/4)=Cb(1+size(Cb,2)/2+...  k\*size(Cwb,2)/4:size(Cb,2)/2+(k+1)\*size(Cwb,2)/4)+...  b\*Cwb(1+size(Cwb,2)/2:3\*size(Cwb,2)/4);  Cb(1+3\*size(Cb,2)/4+k\*size(Cwb,2)/4:3\*size(Cb,2)/4+...  (k+1)\*size(Cwb,2)/4)=Cb(1+3\*size(Cb,2)/4+...  k\*size(Cwb,2)/4:3\*size(Cb,2)/4+(k+1)\*size(Cwb,2)/4)+...  b\*Cwb(1+3\*size(Cwb,2)/4:size(Cwb,2));  k=k+1;  end;  Cb(1:size(Cwb,2)/4)=Cb(1:size(Cwb,2)/4)+b\*Cwb(1:size(Cwb,2)/4);  %% image reconstruction  imr=WAVEREC2(Cr,Sr,'haar');  img=WAVEREC2(Cg,Sg,'haar');  imb=WAVEREC2(Cb,Sb,'haar');  imsize=size(imr);  FAO=zeros(imsize(1),imsize(2),3);  for i=1:imsize(1);  for j=1:imsize(2);  FAO(i,j,1)=imr(i,j);  FAO(i,j,2)=img(i,j);  FAO(i,j,3)=imb(i,j);  end  end  figure, imshow(FAO); title('watermarked image');  **三、**  代码：  syms t w;  f1t=exp(-2\*abs(t));  F1w=fourier(f1t,t,w)  F2w=1/(1+w\*w);  f2t=ifourier(F2w,w,t)  结果：    **四、**  syms t w;  f1t=1/2\*exp(-2\*t)\*heaviside(t);  F1w=fourier(f1t,t,w)  f2t=1/2\*exp(-2\*(t-1))\*heaviside(t-1);  F2w=fourier(f2t,t,w)  F3w=F1w/F2w;  X1=-50;  X2=50;  x1=X1;  dX=0.1;  for a=1:(X2-X1)/dX+1  f1(a)=subs(abs(F1w),w,x1);  f2(a)=subs(abs(F2w),w,x1);  f3(a)=subs(abs(F3w),w,x1);  y1(a)=subs(angle(F1w),w,x1);  y2(a)=subs(angle(F2w),w,x1);  y3(a)=subs(angle(F3w),w,x1);  x1=x1+dX;  end  x=X1:dX:X2;  figure(1)  plot(x,f1)  xlabel('w');  ylabel('|F1w|');  figure(2)  plot(x,f2)  xlabel('w');  ylabel('|F2w|');  figure(3)  plot(x,y1)  xlabel('w');  ylabel('¦È1£¨w£©');  figure(4)  plot(x,y2)  xlabel('w');  ylabel('¦È2£¨w£©');  figure(5)  plot(x,f3)  xlabel('w');  ylabel('|F1w/F2w|');  figure(6)  plot(x,y3)  xlabel('w');  ylabel('¦È(F1w/F2w)'); |
| **实验总结**  通过本次实验，我学会了如何使用MATLAB对图像进行傅里叶变换与反变换，提取高低频分量，以及对高低频进行整合等。也学会了对图片加水印的方式，理解了傅里叶变换在实际生活中的巧妙应用。 |
| **参考文献**  **https://www.jiamisoft.com/blog/21438-szsy.html**  **https://xiongyiming.blog.csdn.net/article/details/89452123?utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-5.control&**  **depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFrom**  **MachineLearnPai2%7Edefault-5.control** |