于2024上半年写出

XXXXXXXXXXXXXXXXXX

姓名：

学号：

专业班级：

中国石油大学（北京）

信息科学与工程学院学生实践类课程学习日志

学生姓名： 学号： 班级：

课程名称：信号处理程序设计

|  |  |  |
| --- | --- | --- |
| **日期** | **主要问题和解决方案** | **指导教师** |
| 5.17 |  |  |
| 5.18 |  |  |
| 5.19 |  |  |
| 5.22 |  |  |
| 5.23 |  |  |
| 5.24 |  |  |
| 5.25 |  |  |
| 5.26 |  |  |
| 5.29 |  |  |
| 5.30 |  |  |
| 5.31 |  |  |

多尺度图像融和

**摘要**：获取图像可以采用不同波段获取图像。红外光图像可以根据热辐射的差异区分目标和背景，这在全天/夜间和所有天气条件下都能很好地工作。相比之下，可见图像可以以与人类视觉系统一致的方式提供具有高空间分辨率和清晰度的纹理细节，本题目旨在将两图像融合，使得最终融合图像同时保持红外图像中的热辐射和可见图像中的纹理。基于多尺度变换的像素级图像融合是计算机视觉领域的研究热点,广泛应用于图像处理领域。

**Abstract**：The acquisition of images can utilize different wavelengths. Infrared image can differentiate targets from backgrounds based on the differences in thermal radiation, which works well in all-day/night and all weather conditions. In contrast, visible images can provide texture details with high spatial resolution and clarity, consistent with the human visual system. The purpose of this topic is to fuse the two images so that the final fused image maintains both the thermal radiation in the infrared image and the texture in the visible image. Pixel-level image fusion based on multi-scale transformation is a research hotspot in the field of computer vision and is widely used in the field of image processing.

**关键词**：图像融合，多尺度变换，红外图像、全天候、可见图像。

**Keywords**：Image fusion, multi-scale transformation, infrared image, all-weather, visible image.

**研究目标**

用小波变换和高斯-拉普拉斯金字塔进行多尺度分解实现图像融合，比较不同小波基对图像融合效果的影响，比较常用的小波基如Daubechies、Haar、Symlets等，并与用金字塔方法进行对比，分析其在图像分解和重构过程中的优势和不足。探索红外与可见光图像融合，将红外图像的热信息与可见光图像的细节信息进行融合，获得更清晰的图像信息。希望将该算法应用于多光谱图像、红外与可见光图像、多视角图像等不同场景，并结合深度学习技术进行优化，为图像处理和分析提供更有效的方法和理论支持。

1. 小波变换

原理：小波变换将图像分解成不同尺度的子带，每个子带对应图像的不同频率信息。高频子带包含图像的细节信息，低频子带包含图像的整体信息。通过选择合适的分解层数，可以根据需要提取不同尺度的特征信息。在每个尺度上，根据选择的融合规则，对来自不同源图像的子带进行融合。最后将融合后的各个尺度子带进行重构，得到最终的融合图像。



实现代码：

% 小波变换实现红外与可见光图像融合

clc;

clear ;

close ;

% 读取红外图像

IR = double(imread('your own path\\IR\\IR.bmp'));

% 读取可见光图像

VI = double(imread('your own path\\VI\\VI.bmp'));

% 对红外和可见光图像进行小波变换,离散小波变换函数dwt2()

%1，2，3，4分别是低频信息，水平方向的高频信息，竖直方向的高频信息，对角线方向的高频信息

[IR1, IR2, IR3, IR4] = dwt2(IR, 'haar');%haar,db1,sym2,sym4。

[VI1, VI2, VI3, VI4] = dwt2(VI, 'haar');

% 对小波系数进行融合%方法一，平均融合方法

fusion1=(IR1 + VI1)/2;%图像的低频信息，刻画原始图像的逼近信息

fusion2=(IR2 + VI2)/2;%图像水平方向的高频信息，刻画原始图像的横向细节

fusion3=(IR3 + VI3)/2;%图像竖直方向的高频信息，刻画原始图像的垂直细节

fusion4=(IR4 + VI4)/2;%图像在对角线方向的高频信息，刻画原始图像的对角线上的细节

% 对融合后的小波系数进行逆变换

blendImage = idwt2(fusion1, fusion2, fusion3, fusion4, 'haar');

% 显示红外和可见光和融合图像

figure(1);

subplot(1, 3, 1);

imshow(uint8(IR));%图像是uint8类型

title('红外光图像');

subplot(1, 3, 2);

imshow(uint8(VI));

title('可见光图像');

subplot(1, 3, 3);

imshow(uint8(blendImage));

title('最终融合图像');

% 保存融合后的图像

imwrite(uint8(blendImage), 'waveletfusion.bmp');

方法2，选择高频能量较大的部分

c1 = max(abs(IR1), abs(VI1));

c2 = max(abs(IR2), abs(VI2));

c3 = max(abs(IR3), abs(VI3));

c4 = max(abs(IR4), abs(VI4));

blendImagesym4 = idwt2(c1, c2, c3, c4, 'haar');%2

figure(2);

imshow(uint8(blendImagesym4));

title('能量较大算法融合图像');

方法3，傅里叶变换图像融合

先写该方法的算法：

function fused\_image = imageFusionFFT\_color(image1, image2)

% 获取图像大小

[rows, cols, ~] = size(image1);

% 对每个颜色通道进行傅里叶变换

for channel = 1:3

fft\_image1\_channel = fft2(image1(:,:,channel));

fft\_image2\_channel = fft2(image2(:,:,channel));

% 计算频谱

spectrum\_image1\_channel = abs(fft\_image1\_channel);

spectrum\_image2\_channel = abs(fft\_image2\_channel);

[a1, b1, c1] = size(spectrum\_image1\_channel);

mask =double(ones(a1, b1, c1)\*0.5);

% 合并两个图像的频谱

fused\_spectrum\_channel = mask .\* fft\_image1\_channel + mask .\* fft\_image2\_channel;

% 进行逆傅里叶变换

fused\_image\_channel = ifft2(fused\_spectrum\_channel);

% 将结果存储到融合图像的对应通道

fused\_image(:,:,channel) = uint8(real(fused\_image\_channel));

end

end

主函数里调用该算法

%第三种方法，傅里叶变换融合方法

fused\_IR1 = imageFusionFFT\_color(IR1, VI1);

fused\_IR2 = imageFusionFFT\_color(IR2, VI2);

fused\_IR3 = imageFusionFFT\_color(IR3, VI3);

fused\_IR4 = imageFusionFFT\_color(IR4, VI4);

fused\_image = idwt2(fused\_IR1, fused\_IR2, fused\_IR3, fused\_IR4, 'haar');

figure(4);

imshow(uint8(fused\_image));

title('傅里叶方法融合图像');

第四种方法，区域能量融合算法

先写该方法的算法：

function new=fenergyy(a,b)

%区域能量算法

% 分解为 RGB 三个通道

R1 = a(:,:,1);

G1 = a(:,:,2);

B1 = a(:,:,3);

R2 = b(:,:,1);

G2 = b(:,:,2);

B2 = b(:,:,3);

% 计算每个通道的区域能量

temp\_ar = nlfilter(R1,[3 3],@nengliang);

temp\_br = nlfilter(R2,[3 3],@nengliang);

temp\_ag = nlfilter(G1,[3 3],@nengliang);

temp\_bg = nlfilter(G2,[3 3],@nengliang);

temp\_ab = nlfilter(B1,[3 3],@nengliang);

temp\_bb = nlfilter(B2,[3 3],@nengliang);

% 根据能量大小进行融合

[mr,nr] = size(R1);

[mg,ng] = size(G1);

[mb,nb] = size(B1);

newr = zeros(mr,nr);

newg = zeros(mg,ng);

newb = zeros(mb,nb);

for i=1:mr

for j=1:nr

if temp\_ar(i,j)>=temp\_br(i,j)

newr(i,j)=R1(i,j);

else

newr(i,j)=R2(i,j);

end end end

for i=1:mg

for j=1:ng

if temp\_ag(i,j)>=temp\_bg(i,j)

newg(i,j)=G1(i,j);

else

newg(i,j)=G2(i,j);

end end end

for i=1:mb

for j=1:nb

if temp\_ab(i,j)>=temp\_bb(i,j)

newb(i,j)=B1(i,j);

else

newb(i,j)=B2(i,j);

end end end

newr = newr/255;

newg = newg/255;

newb = newb/255;

% 合并通道

new = cat(3, newr, newg, newb);

end

function c=nengliang(x) %权值函数

A=[2 4 2; 4 8 4; 2 4 2]; %权值矩阵

C=A.\*x; % 计算每个像素的权重能量

c=sum(sum(C)); % 求和得到区域能量

end

主函数里调用该算法

%第四种，区域能量

q1=fenergyy(IR1,VI1);

q2=fenergyy(IR2,VI2);

q3=fenergyy(IR3,VI3);

q4=fenergyy(IR4,VI4);

qqq=idwt2(q1, q2, q3, q4, 'haar');%4

figure(5);

imshow(qqq);

title('区域融合算法融合图像');

**计算图像融合指标算法**

%峰值信噪比

ir1 = imread('your own path\\IR\\IR.bmp');

vi1 = imread('your own path\\VI\\VI.bmp');

a=imread('your own path\\waveletfusion.bmp');

IR\_psnr = psnr(ir1, a);

VI\_psnr = psnr(vi1, a);

disp(['红外光与融合图峰值信噪比: ' num2str(IR\_psnr)]);

disp(['可见光与融合图峰值信噪比: ' num2str(VI\_psnr)]);

%交叉熵

sourceImage1 = imread('your own path\\IR\\IR.bmp');

sourceImage2 = imread('your own path\\VI\\VI.bmp');

aa=imread('your own path\\waveletfusion.bmp');

sourceImagef = aa;

% 使用 histcounts 函数计算每个像素值的出现次数

[counts1, edges1] = histcounts(sourceImage1(:), 256);

[counts2, edges2] = histcounts(sourceImage2(:), 256);

[countsf, edgesf] = histcounts(sourceImagef(:), 256);

prob1 = counts1 / sum(counts1);% 将出现次数转换为概率

prob2 = counts2 / sum(counts2);

probf = countsf / sum(countsf);

CE1 = crossentropy(prob1, probf);% 计算交叉熵

CE2 = crossentropy(prob2, probf);

disp(['红外光与融合图交叉熵: ' num2str(CE1)]);

disp(['可见光与融合图交叉熵: ' num2str(CE2)]);

%结构相似度

aaa1 = imread('your own path\\IR\\IR.bmp');

aaa2 = imread('your own path\\VI\\VI.bmp');

aaa=imread('your own path\\waveletfusion.bmp');

SSIM1 = ssim(aaa1, aaa);

SSIM2 = ssim(aaa2, aaa);

disp(['红外光与融合图结构相似度: ' num2str(SSIM1)]);

disp(['可见光与融合图结构相似度: ' num2str(SSIM2)]);

%计算互信息

% 使用 entropy 函数计算信息熵

aaaa1 = imread('your own path\\IR\\IR.bmp');

aaaa2 = imread('your own path\\VI\\VI.bmp');

aaaa=imread('your own path\\waveletfusion.bmp');

H1 = entropy(aaaa1);

H2 = entropy(aaaa2);

Hf = entropy(aaaa);

HU = H1 + H2 - Hf;% 计算互信息

disp(['红外光信息熵' num2str(H1)]);

disp(['可见光信息熵' num2str(H2)]);

disp(['融合图信息熵' num2str(Hf)]);

disp(['互信息' num2str(HU)]);

二、用金字塔分解进行多尺度分解实现图像融合

原理：金字塔分解类型图像融合中使用的金字塔有两种类型

(1)高斯金字塔：该金字塔捕获每个级别的低频（近似）信息,它是通过应用高斯滤波器并对图像重复下采样而创建的。高斯金字塔是通过对原始图像进行连续的高斯滤波和下采样操作构建的，高斯金字塔是由底部的最大分辨率图像逐次向下采样得到的一系列图像。

(2)拉普拉斯金字塔：该金字塔捕获每个级别的高频（细节）信息,它是通过减去高斯金字塔的相邻层而构建的。拉普拉斯金字塔的每一层都是由高斯金字塔中对应层的图像减去其下一层图像经过上采样后的结果得到的。

融合方法：先将两个源图像都分解为各自的高斯金字塔和拉普拉斯金字塔，在每个尺度上，根据一定的融合规则，将源图像的拉普拉斯金字塔层进行融合，得到融合后的拉普拉斯金字塔。最后从融合后的拉普拉斯金字塔的最高层开始，逐层向下进行重建，得到融合后的图像。重建过程是通过将当前层的拉普拉斯金字塔上采样再与下一层相加来实现的。

高斯金字塔

**高斯下采样**

可见光图像高斯第一层

可见光高斯第二层

可见光高斯第三层

可见光高斯第四层

**高斯下采样**

**高斯下采样**

相减—

高斯上采样

高斯上采样

相减—

相减—

高斯上采样

可见光拉普拉斯金字塔第一层

拉普拉斯

可见光拉普拉斯金字塔第三层

可见光拉普拉斯金字塔第二层

金字塔

高斯金字塔

红外光图像高斯第一层

红外光光高斯第三层

**高斯下采样**

**高斯下采样**

**高斯下采样**

红外光高斯第四层

红外光高斯第二层

相减—

高斯上采样

高斯上采样

相减—

相减—

高斯上采样

红外光拉普拉斯金字塔第一层

拉普拉斯

红外光拉普拉斯金字塔第三层

红外光拉普拉斯金字塔第二层

金字塔

融合图拉普拉斯金字塔构建与重构

红外光拉普拉斯金字塔第一层

可见光拉普拉斯金字塔第一层

红外光拉普拉斯金字塔第二层

**+**

**+**

可见光拉普拉斯金字塔第二层

**+**

可见光拉普拉斯金字塔第三层

红外光拉普拉斯金字塔第三层

融合金字塔第三层

融合金字塔第二层

融合金字塔第一层

融合金字塔第四层

**上采样**

相加

红外光高斯第四层

可见光高斯第四层

**+**

**上采样**

**上采样**

相加

最终融合图片

相加

实现代码：

（1）主函数main

clc;

clear all;

close all;

Imgaaa=double(imread('your own path\\VI\\VI.bmp'));%可见光图

imgredaaa=double(imread('C:your own path\\IR\\IR.bmp'));%红外图

[rows, cols, channels] = size(imgaaa);

p=4;%金字塔高度

Q1= LaplacianPyramid(imgaaa, p);%可见光图片拉普拉斯金字塔构建

Q2= LaplacianPyramid(imgredaaa, p);%红外光图片拉普拉斯金字塔构建

%构建融合金字塔%

blend\_pyramid = cell(1,p);%设融合高斯金字塔有1,2,3,4层

for i = 1:p%构建融合图像的拉普拉斯金字塔

[a, b, c] = size(Q1{i});%获得可见光/红外光拉普拉斯金字塔的每层图像的尺寸

mute{i} =double(ones(a, b, c)\*0.5);%mute是用来将可见光和红外光相加的分别占比一半，用来调整亮度

blend\_pyramid{i} = Q1{i} .\*mute{i} + Q2{i}.\*mute{i} ;%构建融合图像的拉普拉斯金字塔

blend\_pyramidnengliang{i} = fenergyy(Q1{i},Q2{i});

end

%画图

figure(1);

subplot(3, 4, 1);imshow(uint8(Q2{1}));title('红外Laplace金字塔第一层');

subplot(3, 4, 2);imshow(uint8(Q2{2}));title('红外Laplace金字塔第二层');

subplot(3, 4, 3);imshow(uint8(Q2{3}));title('红外Laplace金字塔第三层');

subplot(3, 4, 4);imshow(uint8(Q2{4}));title('红外Laplace金字塔第四层');

subplot(3, 4, 5);imshow(uint8(Q1{1}));title('可见光Laplace金字塔第一层');

subplot(3, 4, 6);imshow(uint8(Q1{2}));title('可见光Laplace金字塔第二层');

subplot(3, 4, 7);imshow(uint8(Q1{3}));title('可见光Laplace金字塔第三层');

subplot(3, 4, 8);imshow(uint8(Q1{4}));title('可见光Laplace金字塔第四层');

subplot(3, 4, 9);imshow(uint8(blend\_pyramid{1}));title('融合Laplace金字塔第一层');

subplot(3, 4, 10);imshow(uint8(blend\_pyramid{2}));title('融合Laplace金字塔第二层');

subplot(3, 4, 11);imshow(uint8(blend\_pyramid{3}));title('融合Laplace金字塔第三层');

subplot(3, 4, 12);imshow(uint8(blend\_pyramid{4}));title('融合Laplace金字塔第四层');

Q2blendImage = LaplacianReconstruct(blend\_pyramid);%拉普拉斯金字塔重建，拉普拉斯金字塔会比高斯金字塔少一层

Q2blendImagenengliang = LaplacianReconstruct(blend\_pyramidnengliang);

figure(3);

imshow(uint8(Q2blendImage));%输出最后融合的图像

title('融合图像');

figure(4);

imshow((Q2blendImagenengliang));%输出最后融合的图像，区域能量

title('区域能量算法融合图像');

imwrite(uint8(Q2blendImage), 'Pyramid.bmp');%保存

imwrite((Q2blendImagenengliang), 'Pyramid2.bmp');%保存，区域能量

（2）拉普拉斯金字塔构建代码

构造拉普拉斯金字塔

先构建高斯金字塔，然后构建拉普拉斯金字塔

function pyramid=LaplacianPyramid(image, p)

h\_filter = [1, 4, 6, 4, 1]\*(1/16);

h\_filter = h\_filter' \* h\_filter;

g\_filter = h\_filter \* 4;

pyramid = cell(p, 1);

handle\_image = image;

for i = 1:p - 1

%高斯金字塔构建

temp = imfilter(handle\_image, h\_filter, 'replicate');

rows = size(temp, 1);%查询 temp 的第一个维度的长度

cols = size(temp, 2);%查询 temp 的第二个维度的长度

temp = temp(1:2:rows, 1:2:cols, :);%从1到rows和1到cols取值，步长为2

origin\_image = handle\_image;

handle\_image = temp;

%拉普拉斯金字塔构建

temp = UpSampling(handle\_image);%上采样，使得上层图像分辨率与下层相等

temp = temp(1:rows, 1:cols, :);

temp = imfilter(temp, g\_filter, 'replicate');

e\_handle\_image = temp;

pyramid{i} = origin\_image - e\_handle\_image;

end

pyramid{p} = handle\_image;

1. 上采样代码

function up\_image=UpSampling(image)%上采样，使得上层图像分辨率与下层相等

[rows, cols, channels] = size(image);

up\_image = double(zeros(rows\*2, cols\*2, channels));

up\_image(1:2:rows\*2, 1:2:cols\*2, :) = image;

1. 拉普拉斯金字塔重构代码

function image=LaplacianReconstruct(laplacian\_pyramid)

h\_filter = [1, 4, 6, 4, 1]\*(1/16);

g\_filter = h\_filter' \* h\_filter \* 4;%高斯滤波

laplacian\_pyramid\_copy = laplacian\_pyramid;

p = length(laplacian\_pyramid\_copy);%获取金字塔高度

for i = flip(2:p)%i从4到2进行循环，即从金字塔最高层(=可见光和红外光的高斯金字塔最高层相加)开始

temp = laplacian\_pyramid\_copy{i};%将该层赋予temp。先将最高层赋予temp

temp= UpSampling(temp);%上采样

rows= size(laplacian\_pyramid\_copy{i-1}, 1);%获得下一层的行数，1代表返回它的行数

cols= size(laplacian\_pyramid\_copy{i-1}, 2);%获得下一层的列数，2代表返回它的列数

temp = temp(1:rows, 1:cols, :);%让temp的行，列数与下一层一致

temp = imfilter(temp, g\_filter, 'replicate');%使用高斯核进行卷积。replicate代表边界之外的输入数值假定为等于最近的数组边界值

laplacian\_pyramid\_copy{i-1} = laplacian\_pyramid\_copy{i-1} + temp;%上一层经过上采样和滤波后与当前层相加

end

image = laplacian\_pyramid\_copy{1};%返回第一次层，即融合完成

计算图像融合指标算法与小波变换题目方法相同

图像融合算法代码

**方法二，高斯金字塔八层，平均融合方法**

p=8;

**方法三，高斯金字塔四层，可见光和红外光按7:3融合**

p=4;

mute1{i} =double(ones(a, b, c)\*0.7);

mute2{i} =double(ones(a, b, c)\*0.3);

blend\_pyramid{i} = Q1{i} .\*mute1{i} + Q2{i}.\*mute2{i};

**方法四，高斯金字塔四层，可见光和红外光按3:7融合**

p=4;

mute1{i} =double(ones(a, b, c)\*0.3);

mute2{i} =double(ones(a, b, c)\*0.7);

blend\_pyramid{i} = Q1{i} .\*mute1{i} + Q2{i}.\*mute2{i};

**方法五，高斯金字塔八层，区域能量图像融合方法**

p=8;

blend\_pyramidnengliang{i} = fenergyy(Q1{i},Q2{i});

**方法六，高斯金字塔四层，区域能量图像融合方法**

p=4;

blend\_pyramid{i} = fenergyy(Q1{i},Q2{i});

**设计结果与分析：**

红外光信息熵6.7353，可见光信息熵7.0819

1. **小波变换方法：**

(1)采用平均融合方法，不同小波：

①haar,sym1：



红外光与融合图峰值信噪比: 17.1025

可见光与融合图峰值信噪比: 17.0756

红外光与融合图交叉熵: 0.028087

可见光与融合图交叉熵: 0.031167

红外光与融合图结构相似度: 0.57202

可见光与融合图结构相似度: 0.61663

融合图信息熵6.6966

互信息7.1206

②sym2,db2：



红外光与融合图峰值信噪比: 17.0925

可见光与融合图峰值信噪比: 17.0855

红外光与融合图交叉熵: 0.028101

可见光与融合图交叉熵: 0.031131

红外光与融合图结构相似度: 0.57251

可见光与融合图结构相似度: 0.61512

融合图信息熵6.6932

互信息7.124

③db4:



红外光与融合图峰值信噪比: 17.0923

可见光与融合图峰值信噪比: 17.0857

红外光与融合图交叉熵: 0.028103

可见光与融合图交叉熵: 0.031129

红外光与融合图结构相似度: 0.57251

可见光与融合图结构相似度: 0.61519

融合图信息熵6.6925

互信息7.1247

④sym4:



红外光与融合图峰值信噪比: 17.0872

可见光与融合图峰值信噪比: 17.0908

红外光与融合图交叉熵: 0.028664

可见光与融合图交叉熵: 0.03383

红外光与融合图结构相似度: 0.56906

可见光与融合图结构相似度: 0.61918

融合图信息熵6.6991

互信息7.1181

(2)都采用haar小波时，不同融合方法：

红外光图象：



可见光图像：



平均融合后的图像：



选择子带能量较大方法：



傅里叶变换方法：



区域能量融合方法：



**图像融合指标**

方法一，平均融合方法

红外光与融合图峰值信噪比: 17.1025

可见光与融合图峰值信噪比: 17.0756

红外光与融合图交叉熵: 0.028087

可见光与融合图交叉熵: 0.031167

红外光与融合图结构相似度: 0.57202

可见光与融合图结构相似度: 0.61663

融合图信息熵6.6966

互信息7.1206

方法二，选择子带能量较大方法

红外光与融合图峰值信噪比: 14.1282

可见光与融合图峰值信噪比: 13.9048

红外光与融合图交叉熵: 0.023394

可见光与融合图交叉熵: 0.03616

红外光与融合图结构相似度: 0.67918

可见光与融合图结构相似度: 0.41334

融合图信息熵7.1168

互信息6.7005

方法三，傅里叶变换方法

红外光与融合图峰值信噪比: 17.1843

可见光与融合图峰值信噪比: 16.8984

红外光与融合图交叉熵: 0.049719

可见光与融合图交叉熵: 0.053729

红外光与融合图结构相似度: 0.57022

可见光与融合图结构相似度: 0.60043

融合图信息熵6.6142

互信息7.2031

方法四，区域能量融合方法

红外光与融合图峰值信噪比: 14.1689

可见光与融合图峰值信噪比: 13.9638

红外光与融合图交叉熵: 0.02339

可见光与融合图交叉熵: 0.033673

红外光与融合图结构相似度: 0.70818

可见光与融合图结构相似度: 0.44104

融合图信息熵7.113

互信息6.7042

1. **用金字塔分解进行多尺度分解实现图像融合**

红外光图象：



可见光图像：



方法一，高斯金字塔四层，平均融合方法：



方法二，高斯金字塔八层，平均融合方法



方法三，高斯金字塔四层，可见光和红外光按7:3融合



方法四，高斯金字塔四层，可见光和红外光按3:7融合



方法五，高斯金字塔八层，区域能量图像融合方法



方法六，高斯金字塔四层，区域能量图像融合方法



**图像融合指标**

**方法一，高斯金字塔四层，平均融合方法**

红外光与融合图峰值信噪比: 17.1025

可见光与融合图峰值信噪比: 17.0756

红外光与融合图交叉熵: 0.028087

可见光与融合图交叉熵: 0.031167

红外光与融合图结构相似度: 0.57202

可见光与融合图结构相似度: 0.61663

融合图信息熵6.6966

互信息7.1206

**方法二，高斯金字塔八层，平均融合方法**

红外光与融合图峰值信噪比: 17.1025

可见光与融合图峰值信噪比: 17.0756

红外光与融合图交叉熵: 0.028087

可见光与融合图交叉熵: 0.031167

红外光与融合图结构相似度: 0.57202

可见光与融合图结构相似度: 0.61663

融合图信息熵6.6966

互信息7.1206

**方法三，高斯金字塔四层，可见光和红外光按7:3融合**

红外光与融合图峰值信噪比: 14.1676

可见光与融合图峰值信噪比: 21.5236

红外光与融合图交叉熵: 0.025807

可见光与融合图交叉熵: 0.028883

红外光与融合图结构相似度: 0.38436

可见光与融合图结构相似度: 0.78538

融合图信息熵6.8786

互信息6.9386

**方法四，高斯金字塔四层，可见光和红外光按3:7融合**

红外光与融合图峰值信噪比: 21.526

可见光与融合图峰值信噪比: 14.1666

红外光与融合图交叉熵: 0.031688

可见光与融合图交叉熵: 0.033543

红外光与融合图结构相似度: 0.76671

可见光与融合图结构相似度: 0.42945

融合图信息熵6.5666

互信息7.2506

**方法五，高斯金字塔八层，区域能量图像融合方法**

红外光与融合图峰值信噪比: 10.9741

可见光与融合图峰值信噪比: 10.2928

红外光与融合图交叉熵: 0.041712

可见光与融合图交叉熵: 0.043263

红外光与融合图结构相似度: 0.5074

可见光与融合图结构相似度: 0.36956

融合图信息熵7.3061

互信息6.5111

**方法六，高斯金字塔四层，区域能量图像融合方法**

红外光与融合图峰值信噪比: 13.6827

可见光与融合图峰值信噪比: 13.4939

红外光与融合图交叉熵: 0.033784

可见光与融合图交叉熵: 0.041114

红外光与融合图结构相似度: 0.66993

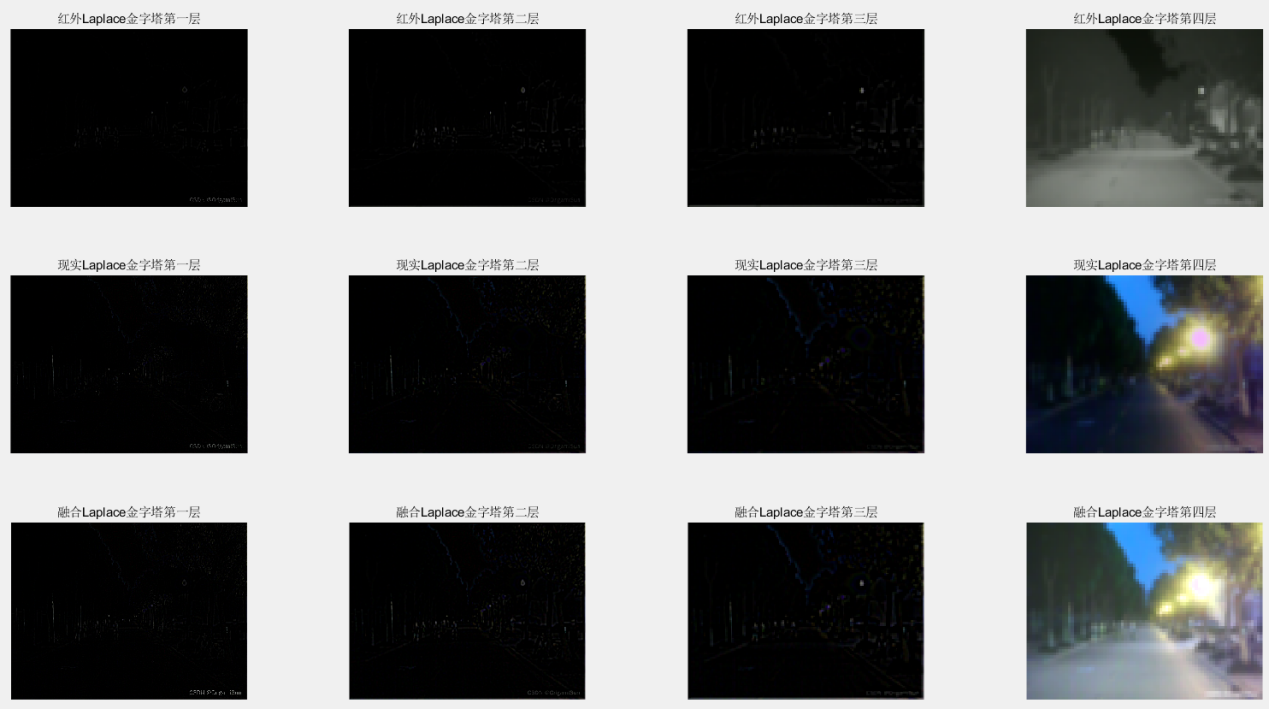
可见光与融合图结构相似度: 0.46094

融合图信息熵7.1508

互信息6.6664

红外、可见光，融合图像的每层拉普拉斯金字塔：(高斯金字塔四层情况）

拉普拉斯金字塔本应该是三层。这里将高斯金字塔最高层作为拉普拉斯金字塔顶层。



分析：

小波变换融合效率高，计算速度快，但是小波变换需要选择合适的小波函数，如Haar,Symlets,Daubechies等，对于不同的图像场景需要进行调整。sym1,haar的图像指标相同,sym2，db2的图像指标相同,可以看出，不同的小波都有各自的优点，采用haar峰值信噪比更高，采用db4互信息更高。傅里叶变换方法和平均融合方法的峰值信噪比都较其他方法高。傅里叶变换方法的互信息最高，但是交叉熵相比其他方法较高，亮度上可以看出选择能量较大方法和区域能量融合算法的亮度更亮。结构相似度上看没有明显优势。

利用高斯-拉普拉斯金字塔进行图像融合可以更好地保留图像的边缘信息，对于细节和纹理的保留效果较好。拉普拉斯金字塔是一种无损的图像金字塔结构。在构建拉普拉斯金字塔时，首先对原始图像进行高斯金字塔分解，然后通过对高斯金字塔不同层次的图像进行差分运算得到拉普拉斯金字塔。这样构建的拉普拉斯金字塔能够保留原始图像的所有信息，并且可以通过逆操作还原回原始图像，实现无损的图像重构，由于拉普拉斯金字塔是通过高斯金字塔的差分运算得到的，可能导致在融合过程中丢失一些细微的图像细节，使得融合结果不够细致，并且拉普拉斯金字塔只能实现固定尺度的图像融合。区域能量图像融合方法在金字塔层数在增多时，会使得图像亮度变高，边缘也变得模糊。高斯金字塔四层，可见光和红外光按3:7融合时互信息最大。可见光和红外光在按照不同比例进行加权融合时候，看到可见光占比越大图片亮度越高，交叉熵越小。而且占比更高的光源类型与融合图像的峰值信噪比更高，结构相似度也更高。平均融合方法金字塔层数不影响图像结果。

参考文献：

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX