

数字图像处理

第四章 Project

中山大学计算机学院 计算机科学与技术

19335174 施天予

本人完成的是 Project04-04 和 Project04-05。

一、高通滤波器 (PROJECT 04-04)

1. 原理

二维高斯滤波器由下式给出

$$H(u, v) = e^{-D^2(u, v)/2\sigma^2}$$

其中 $D(u, v)$ 为距傅里叶变换原点的距离, σ 为高斯曲线扩展的程度。

频率域滤波的流程如下图所示。注意这里要进行图像的**延拓**, 以确保傅里叶变换的正确性。

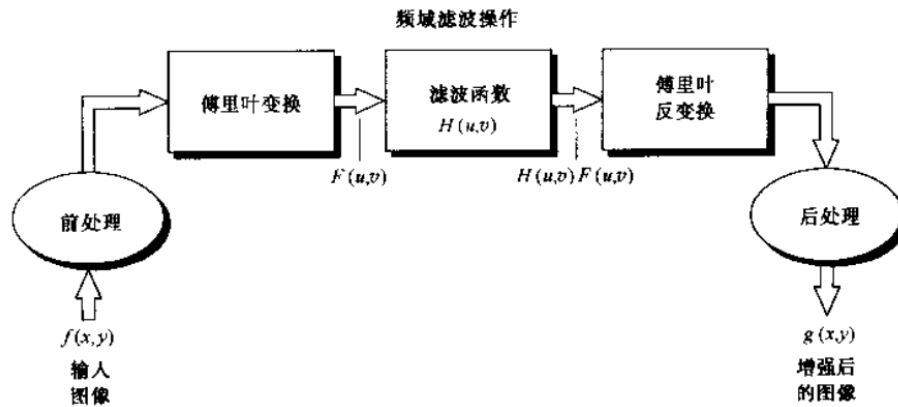


图 1: 频率域滤波的流程

1. 给定大小为 $M \times N$ 的输入图像 $f(x, y)$, 选择填充参数 $P = 2M$, $Q = 2N$
2. 对 $f(x, y)$ 添加必要的 0, 形成大小为 $P \times Q$ 的填充后的图像 $f_p(x, y)$
3. 用 $(-1)^{x+y}$ 乘以 $f_p(x, y)$, 做频谱中心化处理
4. 用上面结果计算 DFT, 得到 $F(u, v)$
5. 生成一个**实对称**的滤波函数 $H(u, v)$, 大小为 $P \times Q$, 中心在 $(P/2, Q/2)$ 处。用阵列相乘得到 $G(u, v) = H(u, v)F(u, v)$
6. 计算上式得到的 IDFT, 并恢复原图像

$$g_p(x, y) = \text{Re}[\Im[G(u, v)](-1)^{x+y}]$$

7. 通过从 $g_p(x, y)$ 的左上角提取 $M \times N$ 大小的区域, 得到最终结果 $g(x, y)$

利用下式可以得到高通滤波的图像

$$f_{hp}(x, y) = f(x, y) - f_{lp}(x, y)$$

其中 f_{lp} 即为 PROJECT 04-03 通过低通滤波后的图像。

2. 实验结果与分析

用原图像减去高斯低通滤波后图像的方法得到的高通图像与直接进行高斯高通滤波的方法产生的图像没有太大区别, 事实上

$$F(u, v) - F(u, v)H(u, v) = F(u, v)(1 - H(u, v)) = F(u, v)G(u, v)$$

其中 H 为高斯低通滤波器, G 为高斯高通滤波器, 因此这两种方式是等价的。故存在差异的原因是数值计算的误差, 即进行傅里叶变换时产生的数值差异导致两幅图像的差异。

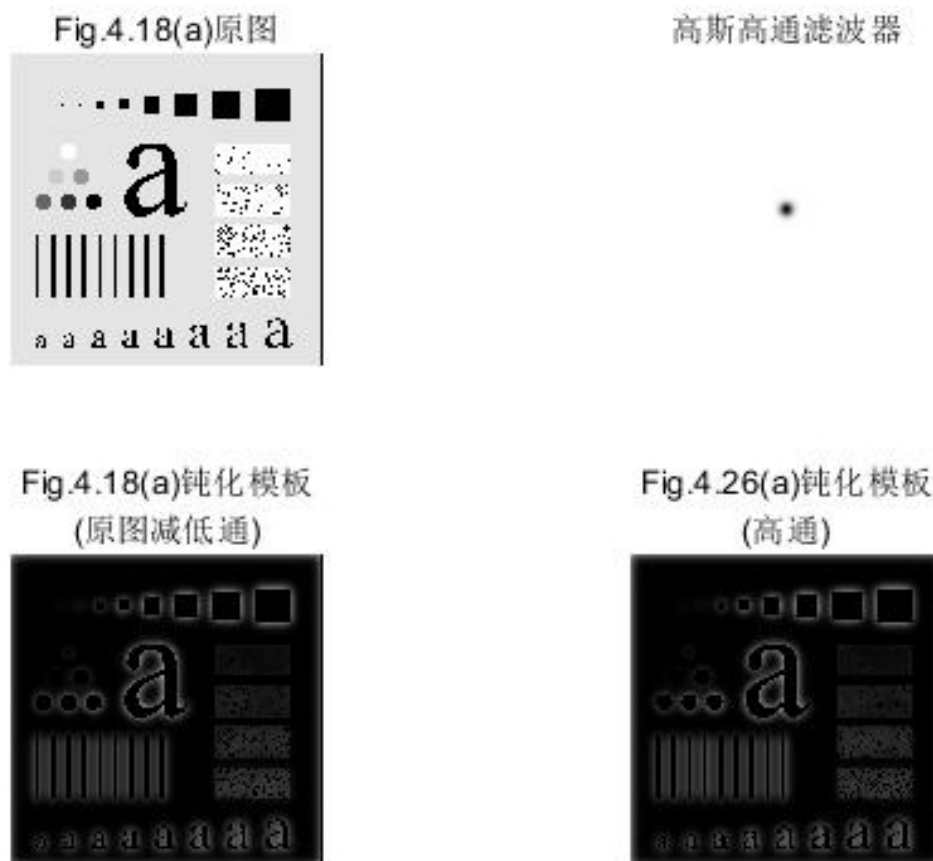


图 2: 高斯高通滤波 $\sigma = 20$

通过加大 σ 的值, 可以使高斯低通滤波器作用的范围更大, 进而剩余的高通分量更少, 也即图像只剩部分细节可见, 图像更加模糊。

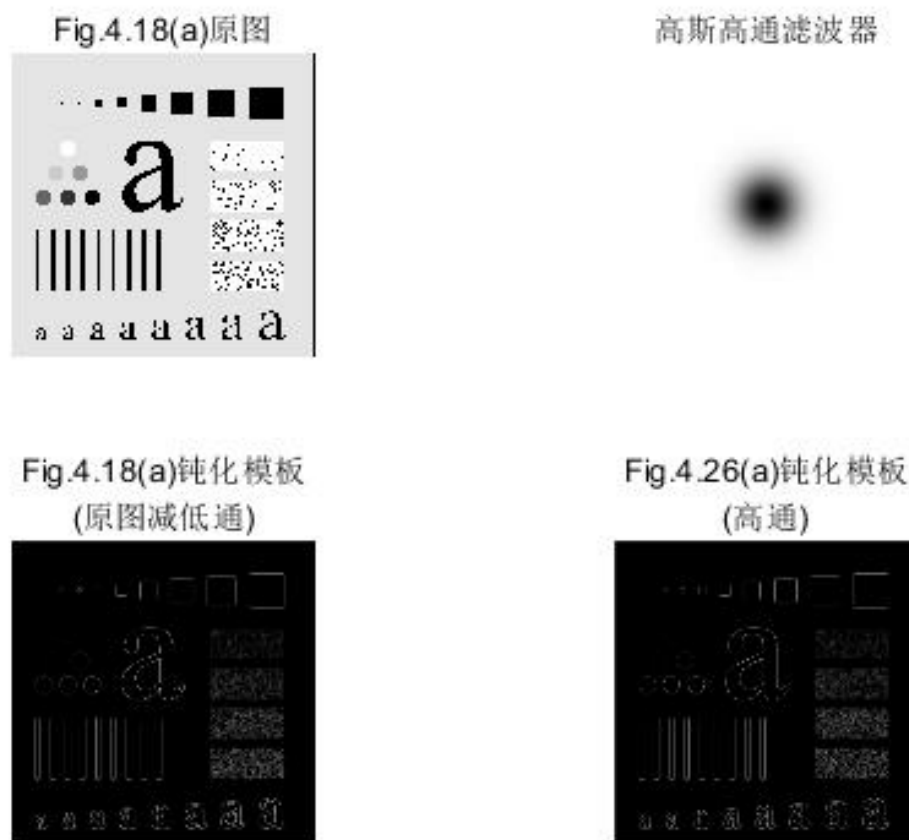


图 3: 高斯高通滤波 $\sigma = 100$

二、频率域的相关 (PROJECT 04-05)

1. 原理

利用空间域的相关与频率域的乘积构成傅里叶变换对这一性质

$$f(x, y) \circ h(x, y) \iff F^*(u, v)H(u, v)$$

程序实现步骤:

1. 将两幅图像的大小都拓展 $298 \times 298 (256 + 42)$
2. 两幅图像的每个像素都乘以 $(-1)^{(x+y)}$, 使其在频域位于中心位置
3. 做傅里叶变换, 转换到频域
4. 在频域两幅图像, 一个与另一个的共轭相乘计算相关函数
5. 作傅里叶逆变换转换回空间域

6. 乘以 $(-1)^{(x+y)}$, 得到最终结果

2. 实验结果与分析

相关性结果如下图所示。

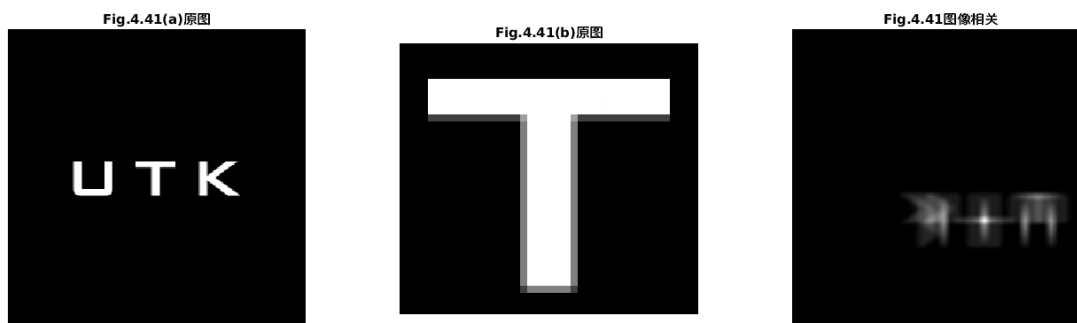


图 4: 频率域的相关

通过 Matlab 计算可以找出最大值出现在 (191,193) 的位置, 符合预期。

附录 A. PROJECT 04-04

```
close all;clc;clear all;
I = imread('Fig0418(a).tif');
I = double(I);
a = 20;
[h,L] = gauss(I,a,1);
l1 = I - h;
[l2,L] = gauss(I,a,0);
figure,
subplot(221),imshow(uint8(I));
title('Fig.4.18(a)原图')
subplot(222),imshow(L);
title('高斯高通滤波器')
subplot(223),imshow(uint8(l1));
title({'Fig.4.18(a)钝化模板','(原图减低通)'})
subplot(224),imshow(uint8(l2));
title({'Fig.4.26(a)钝化模板','(高通)'})

function [g, L] = gauss(img,a,lowpass_flag)
    [M,N] = size(img);
    % 延拓
    P = 2 * M;
    Q = 2 * N;
    % 添加0
```

```

fp = zeros(P,Q);
fp(1:M,1:N) = img(1:M,1:N);
% 频谱中心化处理
[Y,X] = meshgrid(1:Q,1:P);
ones = (-1).^(X+Y);
fp = ones.*fp;
% 傅里叶变换
FLY = fft2(fp);
% 高斯滤波
H = zeros(P,Q);
L = zeros(P,Q);
for i=1:P
    for j=1:Q
        H(i,j) = exp(-(i-M)^2-(j-N)^2)/(2*a^2));
        L(i,j) = 1 - H(i,j);
    end
end
if lowpass_flag == 1
    G = H.*FLY;
else
    G = L.*FLY;
end
% 傅里叶逆变换, 恢复原图像
g = real(ifft2(G));
% 从左上角提取 M*N 大小区域
g = ones.*g;
g = g(1:M,1:N);
end

```

附录 B. PROJECT 04-05

```

close all;clc;clear all;
I1 = imread('Fig0441(a).jpg');
I2 = imread('Fig0441(b).jpg');
[m1,n1] = size(I1);
[m2,n2] = size(I2);
P = 298;
Q = 298;
img1 = zeros(P,Q);
img2 = zeros(P,Q);
img1(1:m1,1:n1) = I1(1:m1,1:n1);
img2(1:m2,1:n2) = I2(1:m2,1:n2);
% 生成网格采样点, 用(-1)^(x+y)乘以原图像
[x,y]=meshgrid(1:P,1:Q);
ones=(-1).^(x+y);

```

```
% 傅里叶变换
f1 = fft2(ones.*img1);
f2 = fft2(ones.*img2);
% 求共轭
rel = f2 .* conj(f1);
% 傅里叶逆变换
newI = ifft2(rel);
newI = ones.*newI;
% 画图
figure,
subplot(131),imshow(I1);
title('Fig.4.41(a)原图')
subplot(132),imshow(I2);
title('Fig.4.41(b)原图')
subplot(133),imshow(mat2gray(newI));
title('Fig.4.41图像相关')
% 二维相关函数中最大值位置的(x,y)坐标
max_value = max(max(newI));
[row,col] = find(newI == max_value);
disp(['max value is : ', num2str(max_value)]);
disp(['row: ', num2str(row), ' col: ', num2str(col)])
```