

# 华中科技大学

## 生物医学数字信号处理实验报告

### 采样和重建

学院	工程科学学院
班级	工程科学学院（生医）1701 班
姓名	汪能志
学号	U201713082

2020 年 4 月 12 日

# 目录

1.	实验内容 .....	1
2.	实验原理 .....	1
2.1.	一维理想采样和重建 .....	1
2.2.	一维非理想采样和重建 .....	2
2.3.	二维采样和重建 .....	3
2.4.	高斯函数和艾里斑 .....	3
2.5.	高斯函数的采样 .....	4
3.	实验内容 .....	5
3.1.	一维高斯函数的理想采样和重建 .....	5
3.2.	一维高斯函数的非理想采样和重建 .....	7
3.3.	二维高斯函数的理想采样和重建 .....	10
3.4.	二维高斯函数的非理想采样和重建 .....	15
3.5.	一维非理想采样的分辨率测试 .....	20
3.6.	二维非理想采样的分辨率测试 .....	23
4.	实验结果 .....	26
4.1.	一维高斯函数的采样和重建 .....	26
4.2.	二维高斯函数的采样和重建 .....	27
4.3.	采样脉冲占空比对采样和重建的影响 .....	30
4.4.	分辨率测试 .....	32
5.	实验分析 .....	33
5.1.	采样周期、采样率和信号重建 .....	33
5.2.	非理想采样和占空比 .....	33
5.3.	空间分辨率 .....	33
5.4.	成像系统的像素尺寸 .....	34
6.	实验心得 .....	34

## 1. 实验内容

对一维高斯形状的信号进行理想采样并恢复；  
对一维高斯形状的信号进行非理想采样并恢复；  
针对二维高斯信号，完成理想采样和非理想采样并恢复；  
根据实验得到的理论和计算结果，指导在成像系统中选择像素的尺寸。

## 2. 实验原理

### 2.1. 一维理想采样和重建

简而言之，对带限连续信号进行理想采样的核心思想是：通过对频域信号的周期延拓，来实现空域（时域）信号的离散化。为了使频域延拓后的信号不产生混叠现象，就需要保证频域延拓的周期不小于连续信号在频域的频率范围。由于实际采集的空域（时域）信号都是实信号，其对应的频域信号满足共轭对称性，因此很容易得到信号的频率范围是其最高频率两倍的结论。因此，对频域信号进行周期延拓的最小周期为空域（时域）信号最高频率的两倍，即 Nyquist 采样定理中对采样率的要求：

$$\omega_s \geq 2\omega_h$$

而从采样信号中重建原始信号的核心，即为将采样得到的离散信号中高于截止频率的部分完全去除。依照实现方法的不同，有频域滤波和空域（时域）内插两种。

#### 2.1.1. 理想采样和频域重建

根据 Nyquist 采样定理，对于带限的模拟信号  $f(x)$ ，如果其最高频率为  $\omega_h$ ，即信号满足：

$$\omega_h = \min \omega, \quad \text{s.t. } |\omega| > \omega_h \quad F(j\omega) = 0$$

则，使用频率  $\omega_s \geq 2\omega_h$  的冲激串进行理想采样后，可以从采样信号中不失真地还原出  $f(x)$ 。此时的理想抽样函数及其傅里叶变换为：

$$p(x) = \sum_{m=-\infty}^{+\infty} \delta(x - mT_s)$$
$$P(j\omega) = \mathcal{F}\{p(x)\} = \frac{2\pi}{T_s} \cdot \sum_{k=-\infty}^{+\infty} \delta(\omega - k\omega_s)$$

则，理想采样后的信号  $f_p(x)$  及其傅里叶变换  $F_p(j\omega)$  为：

$$f_p(x) = f(x) \cdot p(x) = \sum_{n=-\infty}^{+\infty} f(nT_s) \delta(x - nT_s)$$
$$F_p(j\omega) = \mathcal{F}\{f_p(x)\} = \frac{1}{2\pi} F(j\omega) * P(j\omega) = \frac{1}{T_s} \sum_{k=-\infty}^{+\infty} F(j(\omega - k\omega_s))$$

因此，将理想采样后的信号  $f_p(x)$  通过截止频率为  $\omega_s$  的理想低通滤波器  $H(j\omega)$ ，滤除延拓后的多余频谱，就可以不失真地重建连续信号  $f(x)$ 。

$$F_{res}(j\omega) = F_p(j\omega) \cdot H(j\omega)$$

$$f_{res}(j\omega) = \mathcal{F}^{-1}\{F_{res}(j\omega)\} = \mathcal{F}^{-1}\left\{\frac{1}{T_s} \cdot F(j\omega) \cdot T_s\right\} = \mathcal{F}^{-1}\{\mathcal{F}\{f(x)\}\} = f(x)$$

### 2.1.2. 空域内插函数重建

除了在频域滤波再通过傅里叶逆变换重建信号，还可以从空域（或时域）利用卷积重建信号，即为信号重建的内插方法。

理想低通滤波器的冲激响应，即 $H(j\omega)$ 的傅里叶逆变换为：

$$h(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} H(j\omega) \cdot e^{j\omega x} d\omega = \frac{T_s}{2\pi} \int_{-\frac{\omega_s}{2}}^{+\frac{\omega_s}{2}} e^{j\omega x} d\omega = \frac{\sin(\frac{\omega_s t}{2})}{\frac{\omega_s t}{2}} = \frac{\sin(\frac{\pi t}{T_s})}{\frac{\pi t}{T_s}}$$

由傅里叶变换和卷积的联系，有：

$$\begin{aligned} F_{res}(j\omega) &= F_p(j\omega) \cdot H(j\omega) \Leftrightarrow f_{res}(x) = f_p(x) * h(x) \\ f_{res}(x) &= \int_{-\infty}^{+\infty} f_p(\tau) h(x-\tau) d\tau = \int_{-\infty}^{+\infty} f(\tau) \cdot p(\tau) \cdot h(x-\tau) d\tau \\ &= \int_{-\infty}^{+\infty} \left[ \sum_{m=-\infty}^{+\infty} f(\tau) \cdot \delta(\tau - m \cdot T_s) \right] \cdot h(x-\tau) d\tau = \sum_{m=-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(\tau) \cdot \delta(\tau - m \cdot T_s) \cdot h(x-\tau) d\tau \right] \\ &= \sum_{m=-\infty}^{+\infty} f(m \cdot T_s) \cdot h(x - m \cdot T_s) = \sum_{m=-\infty}^{+\infty} f(m \cdot T_s) \cdot \frac{\sin \frac{\pi(x - m \cdot T_s)}{T_s}}{\frac{\pi(x - m \cdot T_s)}{T_s}} \end{aligned}$$

可以得到对带限信号进行理想采样后的内插重建公式。

## 2.2. 一维非理想采样和重建

### 2.2.1. 理想采样和非理想采样的联系

在实际的采样过程中，是无法实现理想采样的。例如现在使用的 CCD 和 CMOS 成像设备，其单个像素都有一定的空间尺寸，其抽样过程实际上是对一定范围内的连续信号进行积分，而非理想采样中要求的周期性冲激串抽样。

另外，理想采样对脉冲序列的要求和实际信号的采集过程在某些时候也存在矛盾。例如，为了更接近理想抽样过程，脉冲序列中的脉冲就需要足够窄，以接近冲激函数。但是，在成像系统在采集图像时，脉冲序列（即单个像素）的面积不能太小，才能保证成像元件有足够的进光量。

### 2.2.2. 采样函数的近似

在实际采样过程中，用于抽样脉冲序列的不是理想的冲激函数序列，而是有一定脉宽的实际采样序列。常见的采样序列有矩形脉冲串，三角脉冲串和sinc函数脉冲串等。由于这里我们需要模拟的是成像过程中面阵感光元件的采样过程，这里我们使用矩形脉冲序列来对其进行近似。

$$p(x) = \begin{cases} \frac{N}{d} & (m \cdot T_s - \frac{d}{2} < x < m \cdot T_s + \frac{d}{2}) \\ 0 & \end{cases}$$

其中 $T_s$ 为采样周期， $d$ 为采样脉冲的宽度，定义 $\frac{d}{T_s}$ 为采样脉冲的占空比。

利用矩形脉冲序列对连续信号进行抽样，并计算得到的抽样信号的频谱。

$$f_p(x) = f(x) \cdot p(x)$$

$$F_p(j\omega) = \frac{1}{T_s} \cdot \sum_{k=-\infty}^{+\infty} \frac{\sin k \frac{\omega_s d}{2N}}{k \frac{\omega_s d}{2N}} \cdot e^{-jkk \frac{\omega_s d}{2N}} \cdot F(j\omega - jk\omega_s)$$

可以得到以下结论，矩形脉冲串采样得到的信号频谱是理想采样的频谱和sinc函数包络后的结果。在采样周期满足 Nyquist 采样定理时，采样信号通过一个低通滤波器后，也可以不失真地重建出原始信号，且采样脉冲的占空比不影响重建结果。

## 2.3. 二维采样和重建

### 2.3.1. 二维采样

根据二维傅里叶变换的可分离性，其可以分离成在两个正交的方向上分别进行一维傅里叶变换。如果信号在 $w$ 和 $v$ 方向上都是带限信号，截止频率为 $w_h$ 和 $v_h$ ，则我们可以把二维采样的过程视为在 $w$ 和 $v$ 方向上，按照各自的采样率 $w_s$ 和 $v_s$ 分别对原始信号的频谱进行周期延拓。

这时，采样脉冲将以 $T_{sx}$ 和 $T_{sy}$ 的空间频率，周期性地分布在 $x-y$ 平面上。

$$p(x, y) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(x - m \cdot T_{sx}, y - n \cdot T_{sy})$$

$$P(jw, jv) = \mathcal{F}\{p(x, y)\} = \frac{(2\pi)^2}{T_{sx} \cdot T_{sy}} \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(w - m \cdot w_s, v - n \cdot v_s)$$

### 2.3.2. 二维重建

将二维采样后的信号，通过在 $w$ 和 $v$ 方向上，截止频率分别为 $w_h$ 和 $v_h$ 的低通滤波器，就可以滤除频域周期延拓后的信号，得到原始的连续信号。这个低通滤波器对应的单位脉冲响应，为 $x$ 和 $y$ 方向上，两个sinc函数的乘积。将单位脉冲响应和原始信号进行卷积，就是连续信号的空域内插重建结果。且和一维重建一样，非理想矩形脉冲采样的占空比不会影响重建结果。

$$h(x, y) = \frac{\sin\left(\pi \cdot \frac{x}{T_{sx}}\right)}{\pi \cdot \frac{x}{T_{sx}}} \cdot \frac{\sin\left(\pi \cdot \frac{y}{T_{sy}}\right)}{\pi \cdot \frac{y}{T_{sy}}}$$

## 2.4. 高斯函数和艾里斑

### 2.4.1. 一维高斯函数及其傅里叶变换

考虑一维高斯函数：

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}}$$

对其做傅里叶变换式：

$$G(j\omega) = \mathcal{F}\{g(x)\} = \int_{-\infty}^{+\infty} g(x) \cdot e^{-j\omega x} dx = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2}{2\sigma^2}} \cdot e^{-j\omega x} dx = \frac{2}{\sqrt{\pi}} \cdot e^{-\frac{1}{2}\omega^2\sigma^2} \cdot \frac{\sqrt{\pi}}{2} = e^{-\frac{\sigma^2\omega^2}{2}}$$

### 2.4.2. 二维高斯函数及其傅里叶变换

考虑二维高斯函数：

$$g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$$

其傅里叶变换为：

$$G(jw, jv) = \mathcal{F}\{g(x, y)\} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y) \cdot e^{-jwx} \cdot e^{-jvy} dx dy = \sqrt{2\pi}\sigma \cdot e^{-\frac{\sigma^2(w^2+v^2)}{2}}$$

可见，一维或二维高斯函数的傅里叶变换式仍为高斯函数。且其标准差是原函数标准差的倒数。这不是一个带限信号，因此不能直接对其进行采样。

### 2.4.3. 艾里斑和高斯函数的一维情况

一般来说，光学成像系统的点扩散函数是一个艾里斑，可以描述为：

$$I(\theta) = I_0 \cdot \left[ \frac{2 \cdot J_1(k \cdot a \cdot \sin \theta)}{k \cdot a \cdot \sin \theta} \right]^2 \quad k \cdot a \cdot \sin \theta = \frac{2\pi a}{\lambda} \cdot \frac{1}{R} \cdot x$$

$$\Delta x \approx 1.22 \cdot \lambda N$$

$\Delta x$  为这个艾里斑的半径为， $\lambda$  和  $N$  是和成像系统有关的参数（波长和焦比），在这里认为是常数。艾里斑可以使用高斯函数来近似描述，以高斯函数和艾里斑尽可能接近为目标，有：

$$I(x) \approx I_0' \cdot e^{-\frac{x^2}{2\sigma^2}} \quad \sigma \approx 0.42 \cdot \lambda N \approx \frac{1}{3} \cdot \Delta x$$

即，我们可以认为一个艾里斑的半径是其使用高斯函数拟合后的标准差的三倍。在讨论光学系统的成像过程时，我们只需要考虑距离高斯函数中心点三倍标准差范围内的信号即可。这一结论同样符合高斯函数的 3- $\sigma$  原则，即高斯函数的大部分能量（99.7%）分布在距其均值三倍标准差的范围内。

### 2.4.4. 艾里斑和高斯函数的二维情况

由于光学成像系统的点扩散函数一般是轴对称的，因此我们认为二维平面上的点扩散函数也可以用二维高斯函数进行拟合。且同样只需要考虑距离高斯函数中心三倍标准差范围内的信号。

## 2.5. 高斯函数的采样

### 2.5.1. 采样范围

由前所述，我们认为在空域中，高斯函数可以近似看做一个有限范围的信号，其分布在距离中心三倍标准差内。

因此，我们只在该范围内进行采样，范围之外的信号认为是 0。

### 2.5.2. 采样率

由前所述，我们认为在空域中，高斯函数可以近似看做一个有限范围的信号，其分布在距离中心三倍标准差内。

相应的，我们可以近似认为高斯函数的频谱也是带限的，其频率范围满足和采样率满足：

$$|\Delta x| \leq 3 \cdot \sigma \quad \omega_h = \frac{3}{\sigma} \quad \omega_s \geq 2 \cdot \omega_h = \frac{6}{\sigma} \quad T_s \leq \frac{2\pi}{\omega_s} = \frac{\pi \cdot \sigma}{3}$$

$$|\Delta x| = |\Delta y| \leq 3 \cdot \sigma w_h = v_h = \frac{3}{\sigma} \quad w_s = v_s \geq 2 \cdot w_h = 2 \cdot v_h = \frac{6}{\sigma} \quad T_{sx} = T_{sy} \leq \frac{2\pi}{w_s} = \frac{2\pi}{v_s} = \frac{\pi \cdot \sigma}{3}$$

### 2.5.3. 实验程序

由于在 MATLAB 程序中不能实现连续信号，因此在实验时，使用一个间隔远小于采样间隔的离散信号来模拟连续信号。同时使用求和来代替积分运算。

## 3. 实验内容

### 3.1. 一维高斯函数的理想采样和重建

```

1  % 一维高斯函数的理想采样和重建
2  %%
3  % 高斯函数标准差
4  sigma = 1;
5  f_sigma = (1 / sigma) / (2 * pi);
6  % 采样率与采样周期
7  Ts = sigma * (pi / 3);
8  Fs = (1 / Ts);
9
10 N = 3001;
11 half_range = 5 * sigma;
12
13 %%
14 % 生成一维高斯信号
15 x = linspace(-half_range, half_range, N);
16 f_x = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
17 Gauss_1D = (1 / (sqrt(2 * pi) * sigma)) * exp(-(x.^2) / (2 * sigma^2));
18 signal_1D = Gauss_1D / max(Gauss_1D(:));
19
20 %%
21 % 采样
22 NT = Ts / ((2 * half_range) / (N - 1));
23 NT = round(NT, 0);
24 sample_num = floor((N - 1) / NT);
25
26 % 确保进行奇数次采样
27 % 最外侧采样时会超出范围，这里忽略。
28 sample_num = sample_num + mod(sample_num - 1, 2) - 2;
29 % 去除三倍标准差外的信号
30 signal_3_sigma = double(abs(x) <= 3 * sigma) .* signal_1D;
31 % 计算位置偏移，确保采样脉冲也关于高斯函数的中心对称
32 center_sample = NT * (sample_num - 1) / 2;
33 center_Gauss = (N - 1) / 2;
34 offset = center_sample - center_Gauss;

```

```

35
36 pulse_sample = zeros(1, N);
37 for i = 1:sample_num
38     center = NT * i - offset;
39     pulse_sample(center) = signal_3_sigma(center);
40 end
41 %%
42 % 计算频谱
43 freq_signal = fft(signal_1D, N);
44 freq_signal = fftshift(freq_signal);
45 freq_signal_amp = abs(freq_signal);
46 freq_signal_amp = freq_signal_amp / max(freq_signal_amp(:));
47
48 freq_sample = fft(pulse_sample, N);
49 freq_sample = fftshift(freq_sample);
50 freq_sample_amp = abs(freq_sample);
51 freq_sample_amp = freq_sample_amp / max(freq_sample_amp(:));
52
53 %%
54 % 频域滤波重建
55 H_recover = double(f_x <= (Fs / 2) & f_x >= -(Fs / 2));
56 freq_recover = H_recover .* freq_sample;
57 freq_recover = ifftshift(freq_recover);
58 signal_recover_freq = abs(ifft(freq_recover));
59 signal_recover_freq = signal_recover_freq / max(signal_recover_freq
    (:));
60
61 %%
62 % 空域内插重建
63 h = sinc(x / Ts);
64 signal_recover_space = conv(pulse_sample, h, 'same');
65 signal_recover_space = signal_recover_space / max(signal_recover_space
    (:));
66
67 %%
68 figure(1)
69 plot(x, signal_1D, 'LineWidth', 2)
70 xlim([-half_range, half_range])
71 ylim([-0.1, 1.1])
72 xticks(-5:1:5)
73 title('一维高斯函数')
74 xlabel('x')
75 ylabel('y')
76 %%
77 figure(2)
78 plot(x, pulse_sample, 'LineWidth', 2)

```



```

79 | xlim([-half_range, half_range])
80 | ylim([-0.1, 1.1])
81 | xticks(-5:1:5)
82 | title('一维高斯函数的理想采样')
83 | xlabel('x')
84 | ylabel('y')
85 | %%
86 | figure(3)
87 | plot(f_x, freq_signal_amp, 'LineWidth', 2)
88 | xlim([-5 * Fs, 5 * Fs])
89 | ylim([-0.1, 1.1])
90 | title('一维高斯函数的归一化频谱')
91 | xlabel('f')
92 | ylabel('F')
93 | %%
94 | figure(4)
95 | plot(f_x, freq_sample_amp, 'LineWidth', 2)
96 | xlim([-5 * Fs, 5 * Fs])
97 | ylim([-0.1, 1.1])
98 | title('一维高斯函数理想采样的归一化频谱')
99 | xlabel('f')
100 | ylabel('F')
101 | %%
102 | figure(5)
103 | plot(x, signal_recover_freq, 'LineWidth', 2)
104 | xlim([-half_range, half_range])
105 | ylim([-0.1, 1.1])
106 | xticks(-5:1:5)
107 | title('理想采样后频域滤波重建结果')
108 | xlabel('x')
109 | ylabel('y')
110 | %%
111 | figure(6)
112 | plot(x, signal_recover_space, 'LineWidth', 2)
113 | xlim([-half_range, half_range])
114 | ylim([-0.1, 1.1])
115 | xticks(-5:1:5)
116 | title('理想采样后空域内插重建结果')
117 | xlabel('x')
118 | ylabel('y')

```

### 3.2. 一维高斯函数的非理想采样和重建

```

1 | % 一维高斯函数的矩形脉冲采样和重建
2 | %%
3 | % 高斯函数标准差
4 | sigma = 1;

```

```

5  f_sigma = (1 / sigma) / (2 * pi);
6  % 采样率与采样周期
7  Ts = sigma * (pi / 3);
8  % Ts = sigma / 6;
9  Fs = (1 / Ts);
10 % 采样脉冲的占空比
11 alpha = 0.25;
12 half_width = 0.5 * alpha * Ts;
13
14 N = 3001;
15 half_range = 5 * sigma;
16
17 %%
18 % 生成一维高斯信号
19 x = linspace(-half_range, half_range, N);
20 f_x = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
21 Gauss_1D = (1 / (sqrt(2 * pi) * sigma)) * exp(-(x.^2) / (2 * sigma^2));
22 signal_1D = Gauss_1D / max(Gauss_1D(:));
23
24 %%
25 % 采样
26 NT = Ts / ((2 * half_range) / (N - 1));
27 NT = round(NT);
28 N_half_width = half_width / ((2 * half_range) / (N - 1));
29 N_half_width = round(N_half_width);
30 sample_num = floor((N - 1) / NT);
31
32 % 确保进行奇数次采样
33 % 最外侧采样时会超出范围，这里忽略。
34 sample_num = sample_num + mod(sample_num - 1, 2) - 2;
35 % 去除三倍标准差外的信号
36 signal_3_sigma = double(abs(x) <= 3 * sigma) .* signal_1D;
37 % 计算位置偏移，确保采样脉冲也关于高斯函数的中心对称
38 center_sample = NT * (sample_num - 1) / 2;
39 center_Gauss = (N - 1) / 2;
40 offset = center_sample - center_Gauss;
41
42 pulse_sample = zeros(1, N);
43 rectangular_sample = zeros(1, N);
44 for i = 1:sample_num
45     center = NT * i - offset;
46     left = center - N_half_width;
47     right = center + N_half_width;
48     Sample_Signal = signal_3_sigma(left:right);
49     Sample_Signal = mean(Sample_Signal, 'all');
50

```

```

51     rectangular_sample(left:right) = Sample_Signal;
52     pulse_sample(center) = Sample_Signal;
53 end
54
55 %%
56 % 计算频谱
57 freq_signal = fft(signal_1D, N);
58 freq_signal = fftshift(freq_signal);
59 freq_signal_amp = abs(freq_signal);
60 freq_signal_amp = freq_signal_amp / max(freq_signal_amp(:));
61
62 freq_sample = fft(rectangular_sample, N);
63 freq_sample = fftshift(freq_sample);
64 freq_sample_amp = abs(freq_sample);
65 freq_sample_amp = freq_sample_amp / max(freq_sample_amp(:));
66
67 %%
68 % 频域滤波重建
69 H_recover = double(f_x <= (Fs / 2) & f_x >= -(Fs / 2));
70 freq_recover = H_recover .* freq_sample;
71 freq_recover = ifftshift(freq_recover);
72 signal_recover_freq = abs(ifft(freq_recover));
73 signal_recover_freq = signal_recover_freq / max(signal_recover_freq
    (:));
74
75 %%
76 % 空域内插重建
77 h = sinc(x / Ts);
78 signal_recover_space = conv(rectangular_sample, h, 'same');
79 signal_recover_space = signal_recover_space / max(signal_recover_space
    (:));
80
81 %%
82 figure(1)
83 plot(x, signal_1D, 'LineWidth', 2)
84 xlim([-half_range, half_range])
85 ylim([-0.1, 1.1])
86 xticks(-5:1:5)
87 title('一维高斯函数')
88 xlabel('x')
89 ylabel('y')
90 %%
91 figure(2)
92 plot(x, rectangular_sample, 'LineWidth', 2)
93 xlim([-half_range, half_range])
94 ylim([-0.1, 1.1])

```

```

95 | xticks(-5:1:5)
96 | title('一维高斯函数的非理想采样')
97 | xlabel('x')
98 | ylabel('y')
99 | %%
100 | figure(3)
101 | plot(f_x, freq_signal_amp, 'LineWidth', 2)
102 | xlim([-5 * Fs, 5 * Fs])
103 | ylim([-0.1, 1.1])
104 | title('一维高斯函数的归一化频谱')
105 | xlabel('f')
106 | ylabel('F')
107 | %%
108 | figure(4)
109 | plot(f_x, freq_sample_amp, 'LineWidth', 2)
110 | xlim([-5 * Fs, 5 * Fs])
111 | ylim([-0.1, 1.1])
112 | title('一维高斯函数非理想采样的归一化频谱')
113 | xlabel('f')
114 | ylabel('F')
115 | %%
116 | figure(5)
117 | plot(x, signal_recover_freq, 'LineWidth', 2)
118 | xlim([-half_range, half_range])
119 | ylim([-0.1, 1.1])
120 | xticks(-5:1:5)
121 | title('非理想采样后频域滤波重建结果')
122 | xlabel('x')
123 | ylabel('y')
124 | %%
125 | figure(6)
126 | plot(x, signal_recover_space, 'LineWidth', 2)
127 | xlim([-half_range, half_range])
128 | ylim([-0.1, 1.1])
129 | xticks(-5:1:5)
130 | title('非理想采样后空域内插重建结果')
131 | xlabel('x')
132 | ylabel('y')

```

### 3.3. 二维高斯函数的理想采样和重建

```

1 | % 二维高斯函数的理想采样和重建
2 | %%
3 | % 高斯函数标准差
4 | sigma = 1;
5 | f_sigma = (1 / sigma) / (2 * pi);
6 |

```

```

7  % 采样率与采样周期
8  Ts = sigma * (pi / 3);
9  Fs = (1 / Ts);
10
11 N = 1501;
12 half_range = 5 * sigma;
13
14 %%
15 % 生成二维高斯信号
16 x = linspace(-half_range, half_range, N);
17 y = linspace(-half_range, half_range, N);
18 [mx, my] = meshgrid(x, y);
19
20 f_x = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
21 f_y = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
22 Gauss_2D = 1 / (sqrt(2 * pi) * sigma) * exp(-(mx.^2 + my.^2) / (2 * sigma^2));
23 signal_2D = Gauss_2D / max(Gauss_2D(:));
24
25 %%
26 % 采样
27 NT = Ts / ((2 * half_range) / (N - 1));
28 NT = round(NT, 0);
29 sample_num = floor((N - 1) / NT);
30
31 % 确保进行奇数次采样
32 % 最外侧采样时会超出范围，这里忽略。
33 sample_num = sample_num + mod(sample_num - 1, 2) - 2;
34 % 去除三倍标准差外的信号
35 Gauss_3_sigma = double(abs(x) <= 3 * sigma & abs(y) <= 3 * sigma) .* signal_2D;
36 % 计算位置偏移，确保采样脉冲也关于高斯函数的中心对称
37 center_sample = NT * (sample_num - 1) / 2;
38 center_Gauss = (N - 1) / 2;
39 offset = center_sample - center_Gauss;
40
41 pulse_sample = zeros(N, N);
42
43 for i = 1:sample_num
44     for j = 1:sample_num
45         center_x = NT * i - offset;
46         center_y = NT * j - offset;
47         pulse_sample(center_x, center_y) = Gauss_3_sigma(center_x, center_y);
48     end
49 end

```

```

50
51 %%
52 % 计算频谱
53 freq_signal = fft2(signal_2D);
54 freq_signal = fftshift(freq_signal);
55 freq_signal_amp = abs(freq_signal);
56 freq_signal_amp = freq_signal_amp / max(freq_signal_amp(:));
57
58 freq_sample = fft2(pulse_sample);
59 freq_sample = fftshift(freq_sample);
60 freq_sample_amp = abs(freq_sample);
61 freq_sample_amp = freq_sample_amp / max(freq_sample_amp(:));
62
63 %%
64 % 频域滤波重建
65 [m_f_x, m_f_y] = meshgrid(f_x, f_y);
66 % 二维理想低通滤波器频率响应
67 H_recover = double(abs(m_f_x) <= (Fs / 2) & abs(m_f_y) <= (Fs / 2));
68 freq_recover = H_recover .* freq_sample;
69 freq_recover = ifftshift(freq_recover);
70
71 freq_recover = ifftshift(freq_recover);
72 signal_recover_freq = abs(ifft2(freq_recover));
73 signal_recover_freq = signal_recover_freq / max(signal_recover_freq
    (:));
74
75 %%
76 % 空域内插重建
77 h = sinc(mx / Ts) .* sinc(my / Ts);
78 signal_recover_space = conv2(pulse_sample, h, 'same');
79 signal_recover_space = signal_recover_space / max(signal_recover_space
    (:));
80
81 %%
82 figure(1)
83 mesh(x, y, signal_2D)
84
85 xlim([-half_range, half_range])
86 ylim([-half_range, half_range])
87 zlim([0, 1])
88 xticks(-5:1:5)
89 yticks(-5:1:5)
90 title('二维高斯函数')
91 xlabel('x')
92 ylabel('y')
93 %%

```

```

94 figure(2)
95 imagesc(x, y, signal_2D)
96 axis equal
97 xlim([-half_range, half_range])
98 ylim([-half_range, half_range])
99 zlim([0, 1])
100 xticks(-5:1:5)
101 yticks(-5:1:5)
102 title('二维高斯函数')
103 xlabel('x')
104 ylabel('y')
105 %%
106 figure(3)
107 mesh(x, y, pulse_sample)
108
109 xlim([-half_range, half_range])
110 ylim([-half_range, half_range])
111 zlim([0, 1])
112 xticks(-5:1:5)
113 yticks(-5:1:5)
114 title('二维高斯函数的理想采样')
115 xlabel('x')
116 ylabel('y')
117 %%
118 figure(4)
119 imagesc(x, y, pulse_sample)
120 axis equal
121 xlim([-half_range, half_range])
122 ylim([-half_range, half_range])
123 zlim([0, 1])
124 xticks(-5:1:5)
125 yticks(-5:1:5)
126 title('二维高斯函数的理想采样')
127 xlabel('x')
128 ylabel('y')
129 %%
130 figure(5)
131 mesh(f_x, f_y, freq_signal_amp)
132
133 xlim([-5 * Fs, 5 * Fs])
134 ylim([-5 * Fs, 5 * Fs])
135 zlim([0, 1])
136 title('二维高斯函数的归一化频谱')
137 xlabel('f_x')
138 ylabel('f_y')
139 %%

```

```

140 figure(6)
141 imagesc(f_x, f_y, freq_signal_amp)
142 axis equal
143 xlim([-5 * Fs, 5 * Fs])
144 ylim([-5 * Fs, 5 * Fs])
145 zlim([0, 1])
146 title('二维高斯函数的归一化频谱')
147 xlabel('f_x')
148 ylabel('f_y')
149 %%
150 figure(7)
151 mesh(f_x, f_y, freq_sample_amp)
152
153 xlim([-5 * Fs, 5 * Fs])
154 ylim([-5 * Fs, 5 * Fs])
155 zlim([0, 1])
156 title('二维高斯函数理想采样的归一化频谱')
157 xlabel('f_x')
158 ylabel('f_y')
159 %%
160 figure(8)
161 imagesc(f_x, f_y, freq_sample_amp)
162 axis equal
163 xlim([-5 * Fs, 5 * Fs])
164 ylim([-5 * Fs, 5 * Fs])
165 zlim([0, 1])
166 title('二维高斯函数理想采样的归一化频谱')
167 xlabel('f_x')
168 ylabel('f_y')
169 %%
170 figure(9)
171 mesh(x, y, signal_recover_freq)
172
173 xlim([-half_range, half_range])
174 ylim([-half_range, half_range])
175 zlim([0, 1])
176 xticks(-5:1:5)
177 yticks(-5:1:5)
178 title('理想采样后频域滤波重建结果')
179 xlabel('x')
180 ylabel('y')
181 %%
182 figure(10)
183 imagesc(x, y, signal_recover_freq)
184 axis equal
185 xlim([-half_range, half_range])

```



```

186 ylim([-half_range, half_range])
187 zlim([0, 1])
188 xticks(-5:1:5)
189 yticks(-5:1:5)
190 title('理想采样后频域滤波重建结果')
191 xlabel('x')
192 ylabel('y')
193 %%
194 figure(11)
195 mesh(x, y, signal_recover_space)
196
197 xlim([-half_range, half_range])
198 ylim([-half_range, half_range])
199 zlim([0, 1])
200 xticks(-5:1:5)
201 yticks(-5:1:5)
202 title('理想采样后空域内插重建结果')
203 xlabel('x')
204 ylabel('y')
205 %%
206 figure(12)
207 imagesc(x, y, signal_recover_space)
208 axis equal
209 xlim([-half_range, half_range])
210 ylim([-half_range, half_range])
211 zlim([0, 1])
212 xticks(-5:1:5)
213 yticks(-5:1:5)
214 title('理想采样后空域内插重建结果')
215 xlabel('x')
216 ylabel('y')

```

### 3.4. 二维高斯函数的非理想采样和重建

```

1 % 二维高斯函数的非理想采样和重建
2 %%
3 % 高斯函数标准差
4 sigma = 1;
5 f_sigma = (1 / sigma) / (2 * pi);
6
7 % 采样率与采样周期
8 Ts = sigma * (pi / 3);
9 Fs = (1 / Ts);
10 % 采样脉冲的占空比
11 alpha = 0.6;
12 half_width = 0.5 * alpha * Ts;
13

```

```

14 N = 1501;
15 half_range = 5 * sigma;
16
17 %%
18 % 生成二维高斯信号
19 x = linspace(-half_range, half_range, N);
20 y = linspace(-half_range, half_range, N);
21 [mx, my] = meshgrid(x, y);
22
23 f_x = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
24 f_y = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
25 Gauss_2D = 1 / (sqrt(2 * pi) * sigma) * exp(-(mx.^2 + my.^2) / (2 * sigma^2));
26 signal_2D = Gauss_2D / max(Gauss_2D(:));
27
28 %%
29 % 采样
30 NT = Ts / ((2 * half_range) / (N - 1));
31 NT = round(NT, 0);
32 N_half_width = half_width / ((2 * half_range) / (N - 1));
33 N_half_width = round(N_half_width);
34 sample_num = floor((N - 1) / NT);
35
36 % 确保进行奇数次采样
37 % 最外侧采样时会超出范围，这里忽略。
38 sample_num = sample_num + mod(sample_num - 1, 2) - 2;
39 % 去除三倍标准差外的信号
40 signal_3_sigma = double(abs(x) <= 3 * sigma & abs(y) <= 3 * sigma) .* signal_2D;
41 % 计算位置偏移，确保采样脉冲也关于高斯函数的中心对称
42 center_sample = NT * (sample_num - 1) / 2;
43 center_Gauss = (N - 1) / 2;
44 offset = center_sample - center_Gauss;
45
46 pulse_sample = zeros(N, N);
47 rectangular_sample = zeros(N, N);
48
49 for i = 1:sample_num
50     for j = 1:sample_num
51         center_x = NT * i - offset;
52         upper=center_x-N_half_width;
53         lower=center_x+N_half_width;
54         center_y = NT * j - offset;
55         left=center_y-N_half_width;
56         right=center_y+N_half_width;
57         Sample_Signal = signal_3_sigma(upper:lower,left:right);

```

```

58     Sample_Signal = mean(Sample_Signal, 'all');
59     rectangular_sample(upper:lower,left:right)=Sample_Signal;
60     pulse_sample(center_x, center_y) =Sample_Signal;
61     end
62 end
63
64 %%
65 % 计算频谱
66 freq_signal = fft2(signal_2D);
67 freq_signal = fftshift(freq_signal);
68 freq_signal_amp = abs(freq_signal);
69 freq_signal_amp = freq_signal_amp / max(freq_signal_amp(:));
70
71 freq_sample = fft2(rectangular_sample);
72 freq_sample = fftshift(freq_sample);
73 freq_sample_amp = abs(freq_sample);
74 freq_sample_amp = freq_sample_amp / max(freq_sample_amp(:));
75
76 %%
77 % 频域滤波重建
78 [m_f_x, m_f_y] = meshgrid(f_x, f_y);
79 % 二维理想低通滤波器频率响应
80 H_recover = double(abs(m_f_x) <= (Fs / 2) & abs(m_f_y) <= (Fs / 2));
81 freq_recover = H_recover .* freq_sample;
82 freq_recover = ifftshift(freq_recover);
83
84 freq_recover = ifftshift(freq_recover);
85 signal_recover_freq = abs(ifft2(freq_recover));
86 signal_recover_freq = signal_recover_freq / max(signal_recover_freq
(:));
87
88 %%
89 % 空域内插重建
90 h = sinc(mx / Ts) .* sinc(my / Ts);
91 signal_recover_space = conv2(rectangular_sample, h, 'same');
92 signal_recover_space = signal_recover_space / max(signal_recover_space
(:));
93
94 %%
95 figure(1)
96 mesh(x, y, signal_2D)
97 xlim([-half_range, half_range])
98 ylim([-half_range, half_range])
99 zlim([0, 1])
100 xticks(-5:1:5)
101 yticks(-5:1:5)

```

```

102 title('二维高斯函数')
103 xlabel('x')
104 ylabel('y')
105 %%
106 figure(2)
107 imagesc(x, y, signal_2D)
108 axis equal
109 xlim([-half_range, half_range])
110 ylim([-half_range, half_range])
111 zlim([0, 1])
112 xticks(-5:1:5)
113 yticks(-5:1:5)
114 title('二维高斯函数')
115 xlabel('x')
116 ylabel('y')
117 %%
118 figure(3)
119 mesh(x, y, rectangular_sample)
120 xlim([-half_range, half_range])
121 ylim([-half_range, half_range])
122 zlim([0, 1])
123 xticks(-5:1:5)
124 yticks(-5:1:5)
125 title('二维高斯函数的非理想采样')
126 xlabel('x')
127 ylabel('y')
128 %%
129 figure(4)
130 imagesc(x, y, rectangular_sample)
131 axis equal
132 xlim([-half_range, half_range])
133 ylim([-half_range, half_range])
134 zlim([0, 1])
135 xticks(-5:1:5)
136 yticks(-5:1:5)
137 title('二维高斯函数的非理想采样')
138 xlabel('x')
139 ylabel('y')
140 %%
141 figure(5)
142 mesh(f_x, f_y, freq_signal_amp)
143
144 xlim([-5 * Fs, 5 * Fs])
145 ylim([-5 * Fs, 5 * Fs])
146 zlim([0, 1])
147 title('二维高斯函数的归一化频谱')

```

```

148 xlabel('f_x')
149 ylabel('f_y')
150 %%
151 figure(6)
152 imagesc(f_x, f_y, freq_signal_amp)
153 axis equal
154 xlim([-5 * Fs, 5 * Fs])
155 ylim([-5 * Fs, 5 * Fs])
156 zlim([0, 1])
157 title('二维高斯函数的归一化频谱')
158 xlabel('f_x')
159 ylabel('f_y')
160 %%
161 figure(7)
162 mesh(f_x, f_y, freq_sample_amp)
163
164 xlim([-5 * Fs, 5 * Fs])
165 ylim([-5 * Fs, 5 * Fs])
166 zlim([0, 1])
167 title('二维高斯函数非理想采样的归一化频谱')
168 xlabel('f_x')
169 ylabel('f_y')
170 %%
171 figure(8)
172 imagesc(f_x, f_y, freq_sample_amp)
173 axis equal
174 xlim([-5 * Fs, 5 * Fs])
175 ylim([-5 * Fs, 5 * Fs])
176 zlim([0, 1])
177 title('二维高斯函数非理想采样的归一化频谱')
178 xlabel('f_x')
179 ylabel('f_y')
180 %%
181 figure(9)
182 mesh(x, y, signal_recover_freq)
183
184 xlim([-half_range, half_range])
185 ylim([-half_range, half_range])
186 zlim([0, 1])
187 xticks(-5:1:5)
188 yticks(-5:1:5)
189 title('非理想采样后频域滤波重建结果')
190 xlabel('x')
191 ylabel('y')
192 %%
193 figure(10)

```

```

194 imagesc(x, y, signal_recover_freq)
195 axis equal
196 xlim([-half_range, half_range])
197 ylim([-half_range, half_range])
198 zlim([0, 1])
199 xticks(-5:1:5)
200 yticks(-5:1:5)
201 title('非理想采样后频域滤波重建结果')
202 xlabel('x')
203 ylabel('y')
204 %%
205 figure(11)
206 mesh(x, y, signal_recover_space)
207
208 xlim([-half_range, half_range])
209 ylim([-half_range, half_range])
210 zlim([0, 1])
211 xticks(-5:1:5)
212 yticks(-5:1:5)
213 title('非理想采样后空域内插重建结果')
214 xlabel('x')
215 ylabel('y')
216 %%
217 figure(12)
218 imagesc(x, y, signal_recover_space)
219 axis equal
220 xlim([-half_range, half_range])
221 ylim([-half_range, half_range])
222 zlim([0, 1])
223 xticks(-5:1:5)
224 yticks(-5:1:5)
225 title('非理想采样后空域内插重建结果')
226 xlabel('x')
227 ylabel('y')

```

### 3.5. 一维非理想采样的分辨率测试

```

1 % 一维分辨率测试
2 %%
3 % 高斯函数标准差
4 sigma = 1;
5 f_sigma = (1 / sigma) / (2 * pi);
6 % 采样率与采样周期
7 Ts = sigma * (pi / 3);
8 Fs = (1 / Ts);
9 % 采样脉冲的占空比
10 alpha = 0.75;

```

```

11 half_width = 0.5 * alpha * Ts;
12
13 N = 10001;
14 half_range = 10 * sigma;
15
16 %%
17 % 生成一维信号
18 point_center = [-1, 0, 1];
19 point_center = 3 .* sigma .* point_center;
20 num_point = size(point_center, 2);
21 x = linspace(-half_range, half_range, N);
22 f_x = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
23
24 Gauss_1D = zeros(1, N, num_point);
25
26 for i = 1:num_point
27     x_c = point_center(i);
28     PSF_1D = (1 / (sqrt(2 * pi) * sigma)) * exp(-((x - x_c).^2) / (2 * sigma^2));
29     % 去除三倍标准差外的信号
30     Gauss_1D(:, :, i) = double(abs(x - x_c) <= 3 * sigma) .* PSF_1D;
31 end
32
33 signal_1D = sum(Gauss_1D, 3);
34 signal_1D = signal_1D / max(signal_1D(:));
35 %%
36 % 采样
37 NT = Ts / ((2 * half_range) / (N - 1));
38 NT = round(NT);
39 N_half_width = half_width / ((2 * half_range) / (N - 1));
40 N_half_width = round(N_half_width);
41 sample_num = floor((N - 1) / NT);
42
43 % 确保进行奇数次采样
44 % 最外侧采样时会超出范围，这里忽略。
45 sample_num = sample_num + mod(sample_num - 1, 2) - 2;
46
47 % 计算位置偏移，确保采样脉冲关于中心对称
48 center_sample = NT * (sample_num - 1) / 2;
49 center_Gauss = (N - 1) / 2;
50 offset = center_sample - center_Gauss;
51
52 pulse_sample = zeros(1, N);
53 rectangular_sample = zeros(1, N);
54 for i = 1:sample_num
55     center = NT * i - offset;

```

```

56     left = center - N_half_width;
57     right = center + N_half_width;
58     Sample_Signal = signal_1D(left:right);
59     Sample_Signal = mean(Sample_Signal, 'all');
60
61     rectangular_sample(left:right) = Sample_Signal;
62     pulse_sample(center) = Sample_Signal;
63 end
64
65 %%
66 % 计算频谱
67 freq_sample = fft(rectangular_sample, N);
68 freq_sample = fftshift(freq_sample);
69 freq_sample_amp = abs(freq_sample);
70 freq_sample_amp = freq_sample_amp / max(freq_sample_amp(:));
71
72 %%
73 % 频域滤波重建
74 H_recover = double(f_x <= (Fs / 2) & f_x >= -(Fs / 2));
75 freq_recover = H_recover .* freq_sample;
76 freq_recover = ifftshift(freq_recover);
77 signal_recover_freq = abs(ifft(freq_recover));
78 signal_recover_freq = signal_recover_freq / max(signal_recover_freq
    (:));
79
80 %%
81 figure(1)
82 plot(x, signal_1D, 'LineWidth', 2)
83 xlim([-half_range, half_range])
84 ylim([-0.1, 1.1])
85 xlabel('x')
86 ylabel('y')
87 title('一维分辨率测试原始信号')
88
89 figure(2)
90 plot(x, rectangular_sample, 'LineWidth', 2)
91 xlim([-half_range, half_range])
92 ylim([-0.1, 1.1])
93 xlabel('x')
94 ylabel('y')
95 title('一维分辨率测试采样信号')
96
97 figure(3)
98 plot(x, signal_recover_freq, 'LineWidth', 2)
99 xlim([-half_range, half_range])
100 ylim([-0.1, 1.1])

```



```

101 xlabel('x')
102 ylabel('y')
103 title('一维分辨率测试重建信号')

```

### 3.6. 二维非理想采样的分辨率测试

```

1  % 二维分辨率测试
2  %%
3  % 高斯函数标准差
4  sigma = 1;
5  f_sigma = (1 / sigma) / (2 * pi);
6  % 采样率与采样周期
7  Ts = sigma * (pi / 3);
8  Fs = (1 / Ts);
9  % 采样脉冲的占空比
10 alpha = 0.9;
11 half_width = 0.5 * alpha * Ts;
12
13 N = 4501;
14 half_range = 9 * sigma;
15 %%
16 % 生成二维信号
17 point_center = [
18     0, 0;
19     cos(0), sin(0);
20     cos(pi / 3), sin(pi / 3);
21     cos(2 * pi / 3), sin(2 * pi / 3);
22     cos(3 * pi / 3), sin(3 * pi / 3);
23     cos(4 * pi / 3), sin(4 * pi / 3);
24     cos(5 * pi / 3), sin(5 * pi / 3);
25 ];
26
27 point_center = 3 .* sigma .* point_center;
28
29 num_point = size(point_center, 1);
30 x = linspace(-half_range, half_range, N);
31 y = linspace(-half_range, half_range, N);
32 [mx, my] = meshgrid(x, y);
33
34 f_x = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
35 f_y = linspace(-(N - 1) / 2, (N - 1) / 2, N) ./ (2 * half_range);
36
37 Gauss_2D = zeros(N, N, num_point);
38
39 for i = 1:num_point
40     x_c = point_center(i, 1);

```

```

41     y_c = point_center(i, 2);
42     PSF_2D = 1 / (sqrt(2 * pi) * sigma) * exp(-((mx - x_c).^2 + (my - y_
c).^2) / (2 * sigma^2));
43     % 去除三倍标准差外的信号
44     Gauss_2D(:, :, i) = double(abs(mx - x_c) <= 3 * sigma & abs(my - y_
c) <= 3 * sigma) .* PSF_2D;
45 end
46
47 signal_2D = sum(Gauss_2D, 3);
48 signal_2D = signal_2D / max(signal_2D(:));
49
50 %%
51 % 采样
52 NT = Ts / ((2 * half_range) / (N - 1));
53 NT = round(NT, 0);
54 N_half_width = half_width / ((2 * half_range) / (N - 1));
55 N_half_width = round(N_half_width);
56 sample_num = floor((N - 1) / NT);
57
58 % 确保进行奇数次采样
59 % 最外侧采样时会超出范围，这里忽略。
60 sample_num = sample_num + mod(sample_num - 1, 2) - 2;
61 % 计算位置偏移，确保采样脉冲关于中心对称
62 center_sample = NT * (sample_num - 1) / 2;
63 center_Gauss = (N - 1) / 2;
64 offset = center_sample - center_Gauss;
65
66 rectangular_sample = zeros(N, N);
67
68 for i = 1:sample_num
69     for j = 1:sample_num
70         center_x = NT * i - offset;
71         upper = center_x - N_half_width;
72         lower = center_x + N_half_width;
73
74         center_y = NT * j - offset;
75         left = center_y - N_half_width;
76         right = center_y + N_half_width;
77
78         Sample_Signal = signal_2D(upper:lower, left:right);
79         Sample_Signal = mean(Sample_Signal, 'all');
80         rectangular_sample(upper:lower, left:right) = Sample_Signal;
81     end
82 end
83
84 %%

```

```

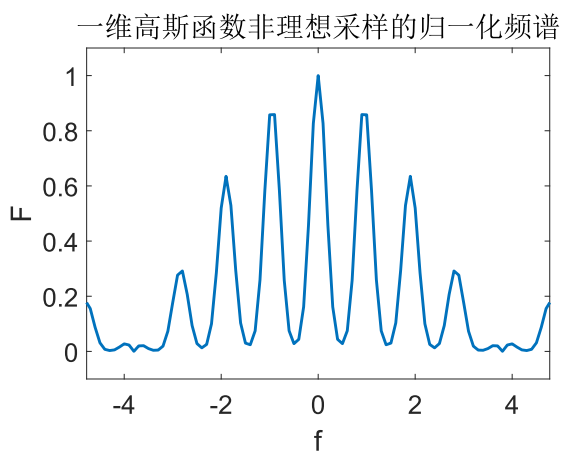
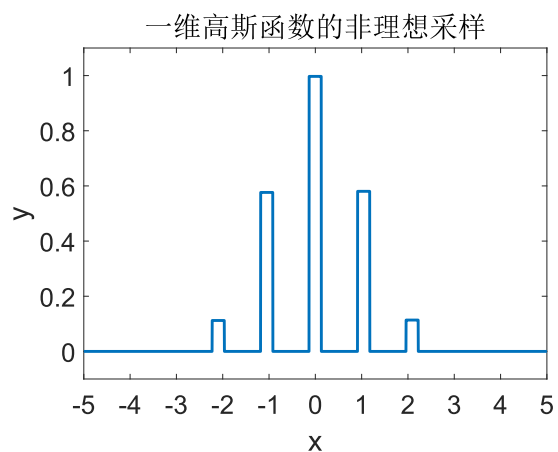
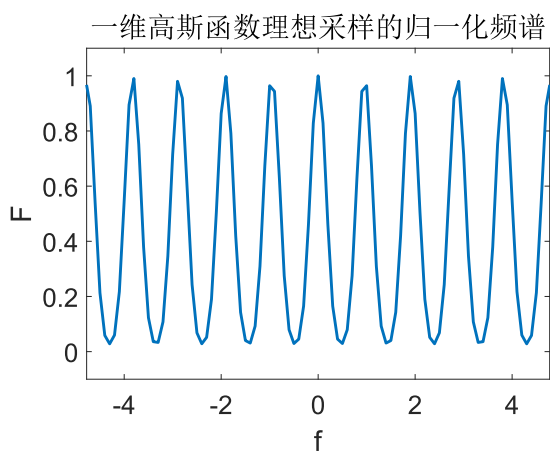
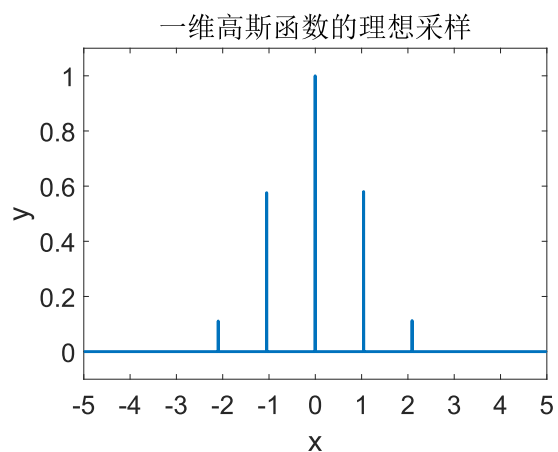
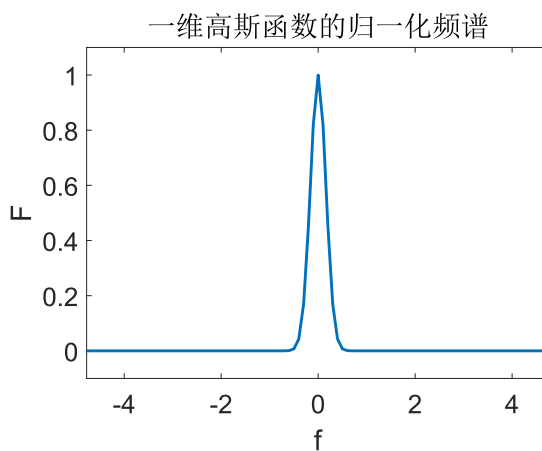
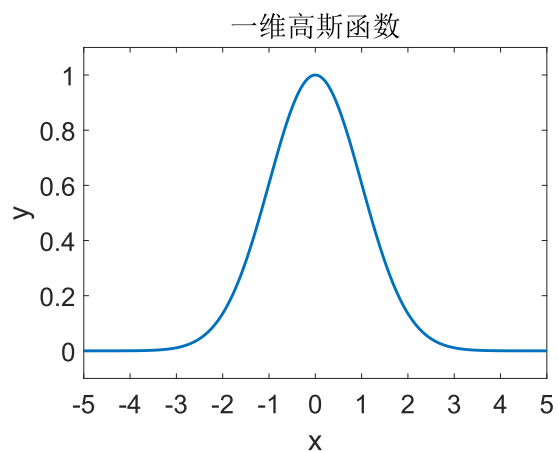
85 % 计算采样信号的频谱
86 freq_sample = fft2(rectangular_sample);
87 freq_sample = fftshift(freq_sample);
88 freq_sample_Amp = abs(freq_sample);
89 freq_sample_Amp = freq_sample_Amp / max(freq_sample_Amp(:));
90
91 %%
92 % 频域滤波重建
93 [m_f_x, m_f_y] = meshgrid(f_x, f_y);
94 % 二维理想低通滤波器频率响应
95 H_recover = double(abs(m_f_x) <= (Fs / 2) & abs(m_f_y) <= (Fs / 2));
96 freq_recover = H_recover .* freq_sample;
97
98 freq_recover = ifftshift(freq_recover);
99 signal_recover = abs(ifft2(freq_recover));
100 signal_recover = signal_recover / max(signal_recover(:));
101
102 %%
103 figure(1)
104 imagesc(x, y, signal_2D)
105 axis equal
106 xlim([-6, 6])
107 ylim([-6, 6])
108 xlabel('x')
109 ylabel('y')
110 title('二维分辨率测试原始信号')
111
112 figure(2)
113 imagesc(x, y, rectangular_sample)
114 axis equal
115 xlim([-6, 6])
116 ylim([-6, 6])
117 xlabel('x')
118 ylabel('y')
119 title('二维分辨率测试采样信号')
120
121 figure(3)
122 imagesc(x, y, signal_recover)
123 axis equal
124 xlim([-6, 6])
125 ylim([-6, 6])
126 xlabel('x')
127 ylabel('y')
128 title('二维分辨率测试重建信号')

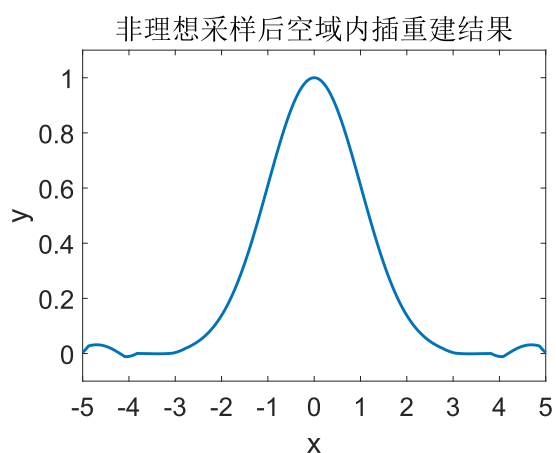
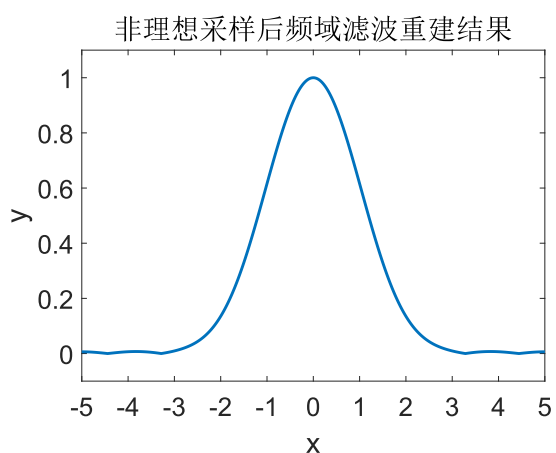
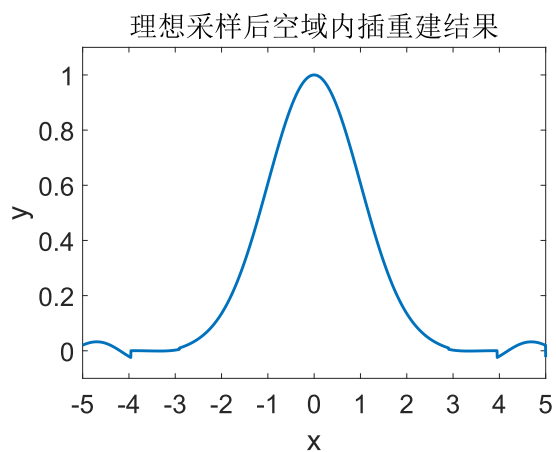
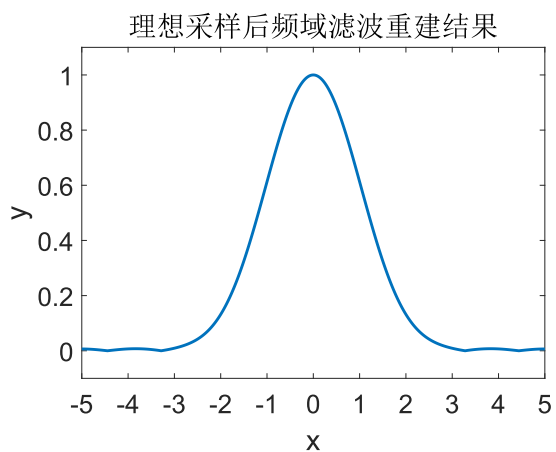
```

## 4. 实验结果

### 4.1. 一维高斯函数的采样和重建

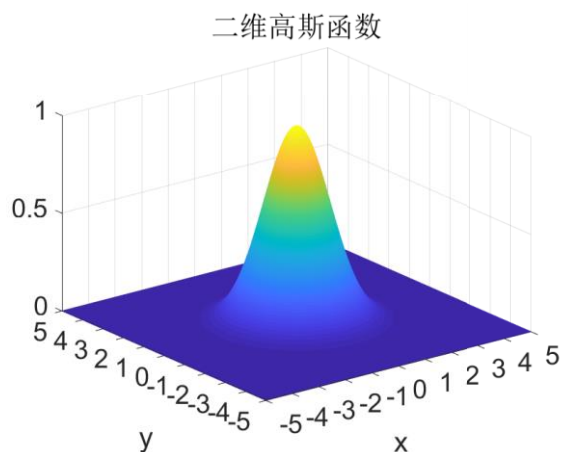
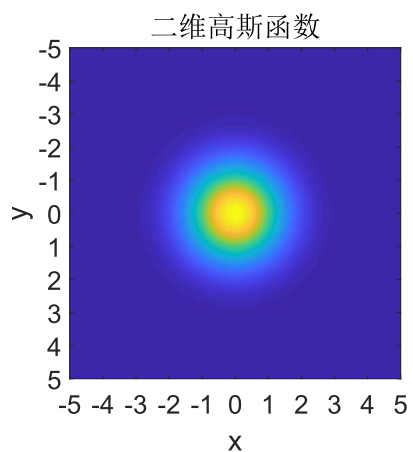
非理想采样中采样脉冲宽度的占空比为 0.25。



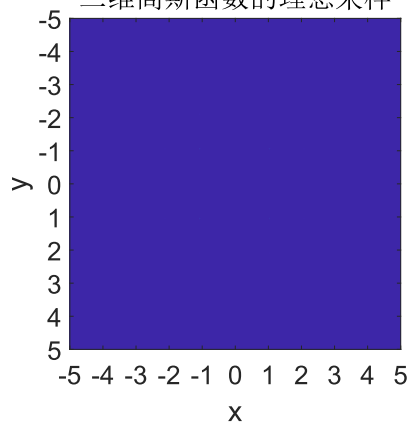


## 4.2. 二维高斯函数的采样和重建

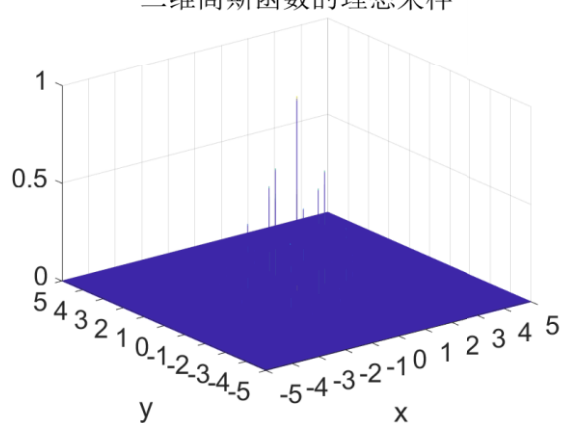
非理想采样中采样脉冲宽度的占空比为 0.6。



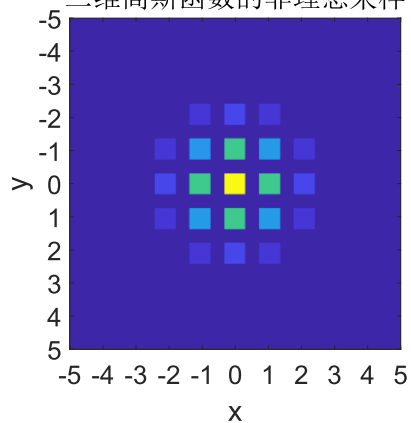
二维高斯函数的理想采样



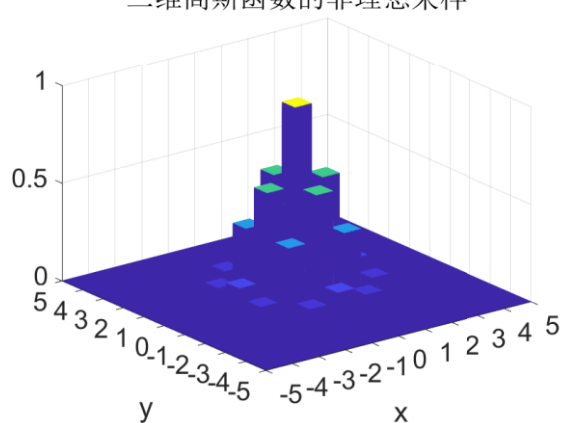
二维高斯函数的理想采样



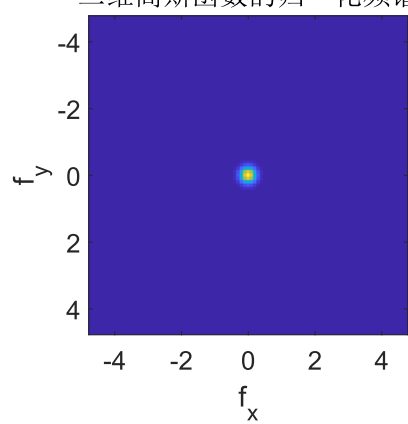
二维高斯函数的非理想采样



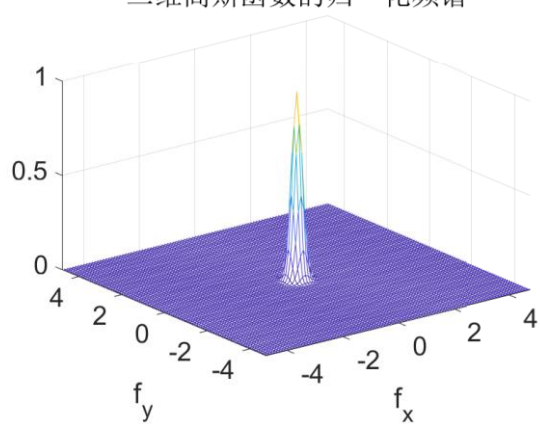
二维高斯函数的非理想采样



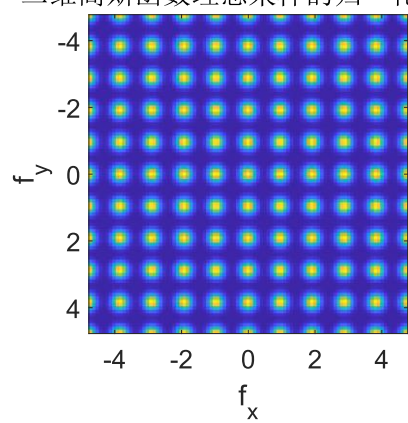
二维高斯函数的归一化频谱



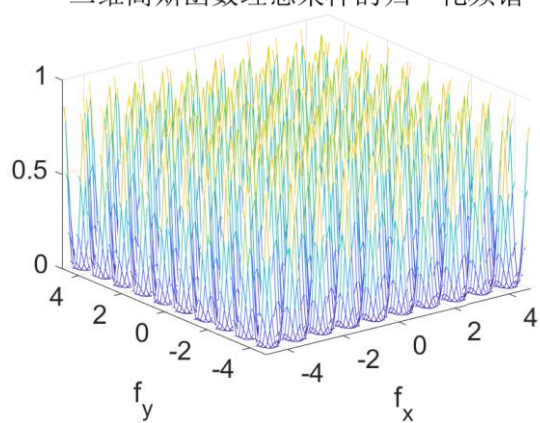
二维高斯函数的归一化频谱



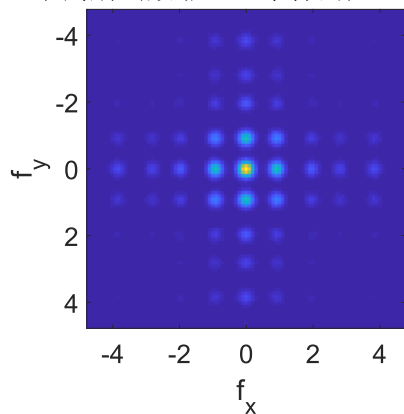
二维高斯函数理想采样的归一化频谱



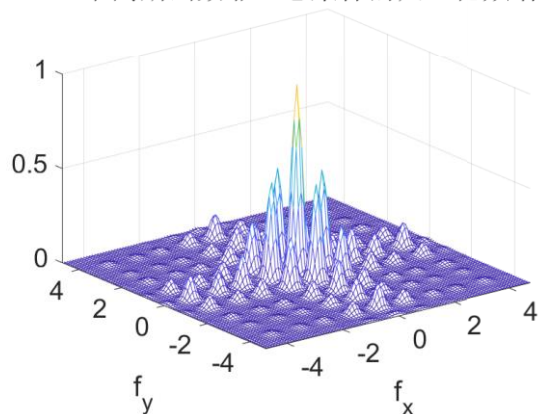
二维高斯函数理想采样的归一化频谱



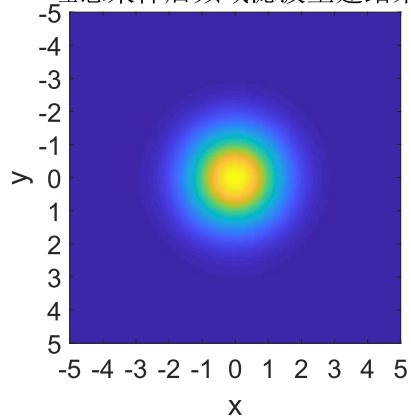
二维高斯函数非理想采样的归一化频谱



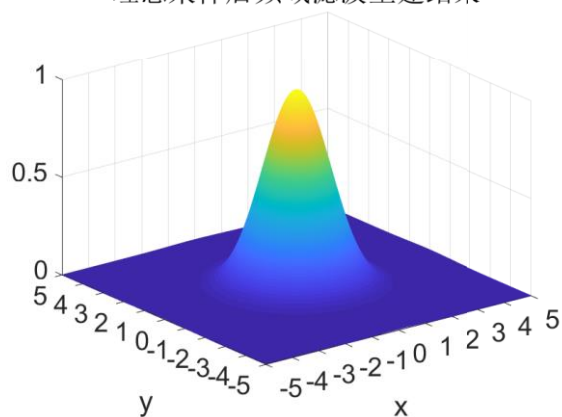
二维高斯函数非理想采样的归一化频谱



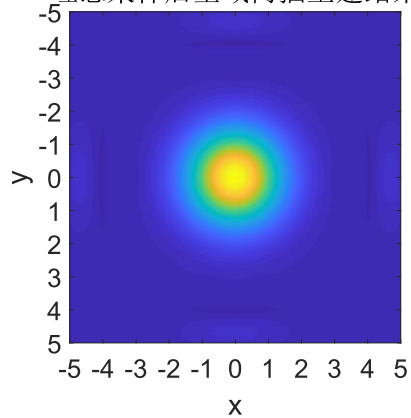
理想采样后频域滤波重建结果



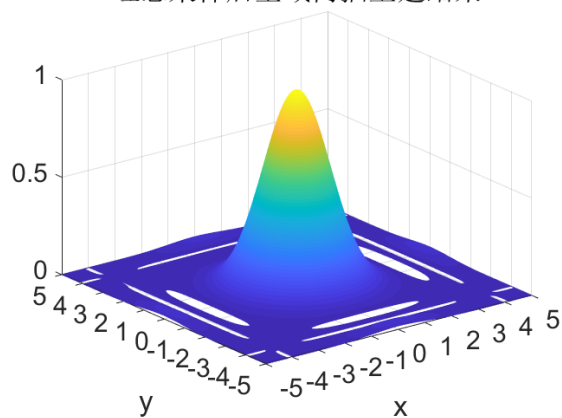
理想采样后频域滤波重建结果



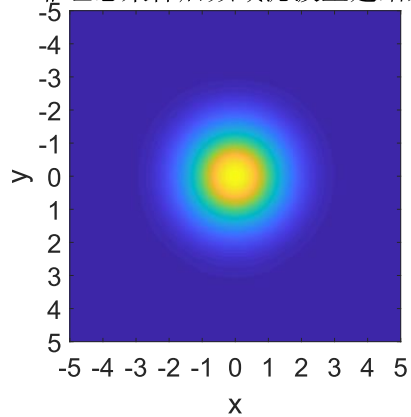
理想采样后空域内插重建结果



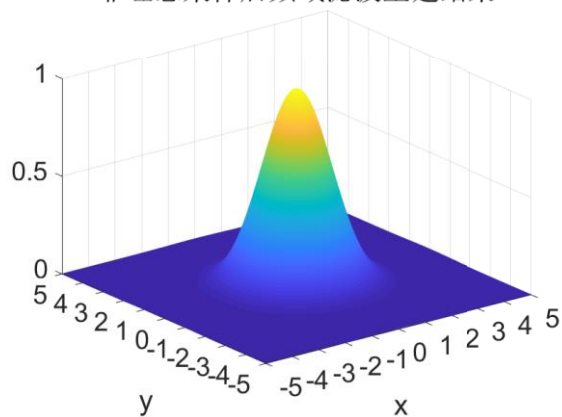
理想采样后空域内插重建结果

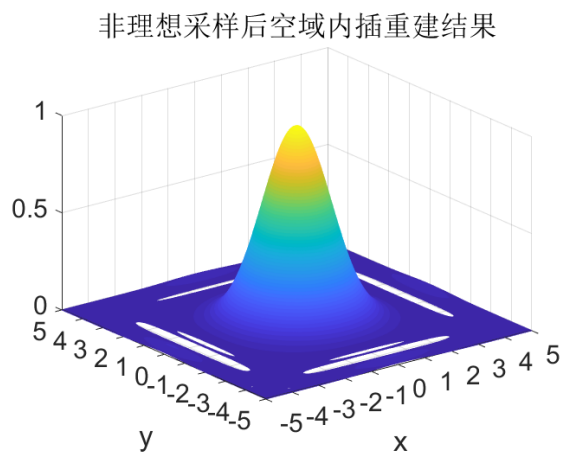
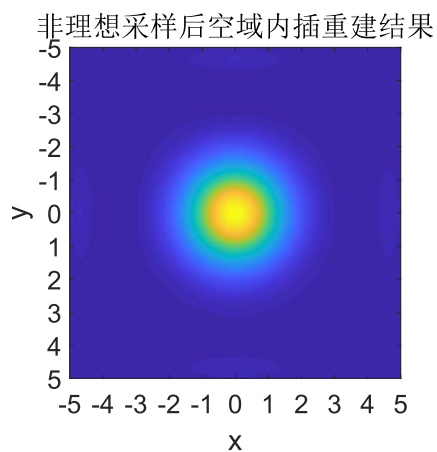


非理想采样后频域滤波重建结果

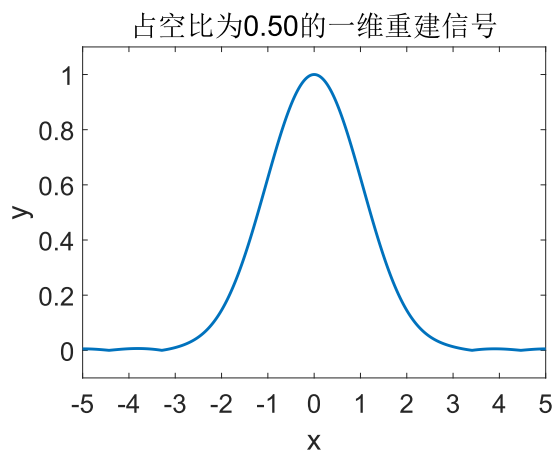
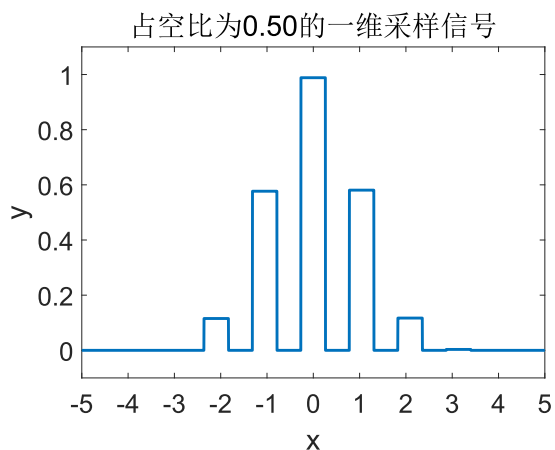
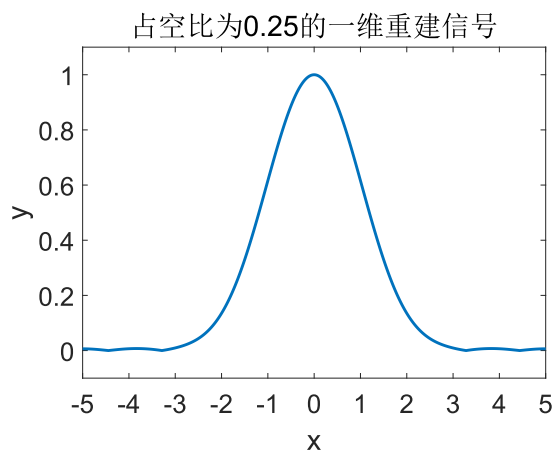
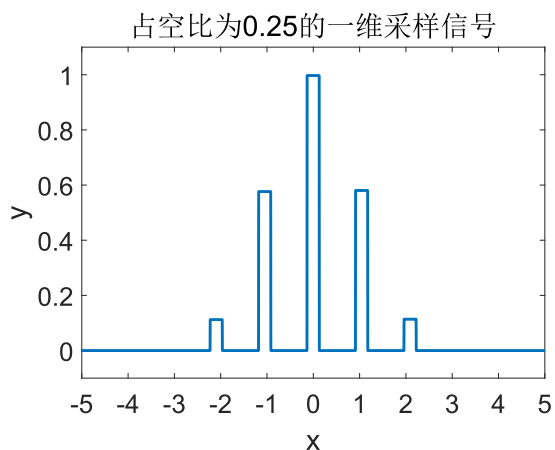
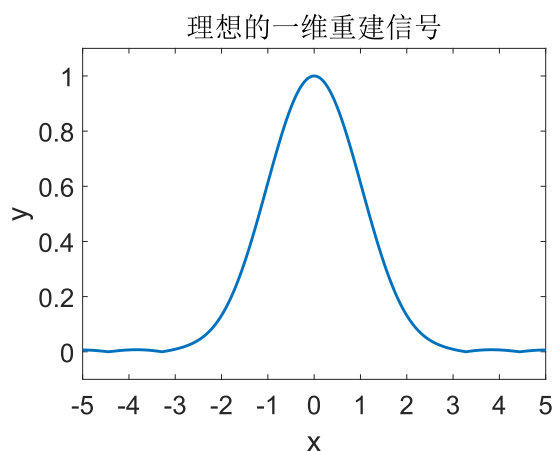
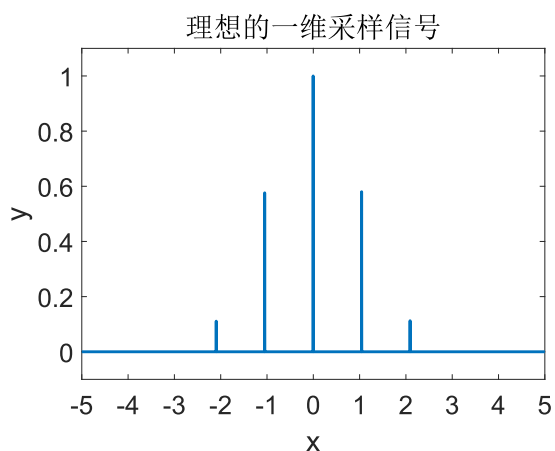


非理想采样后频域滤波重建结果

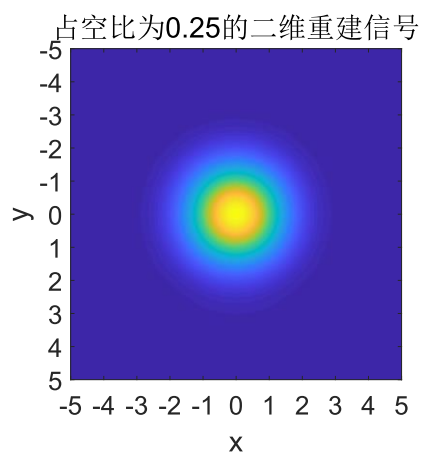
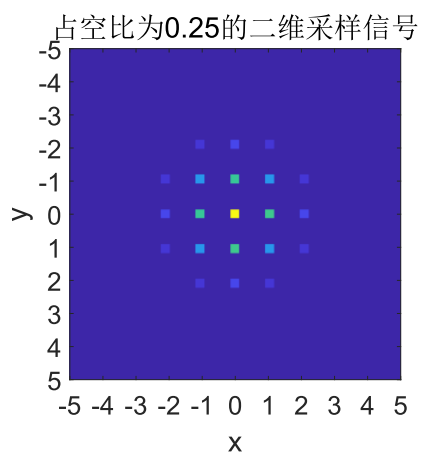
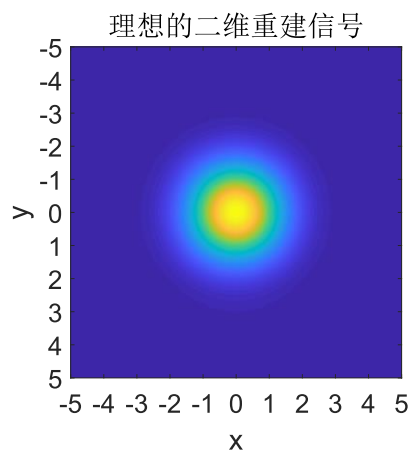
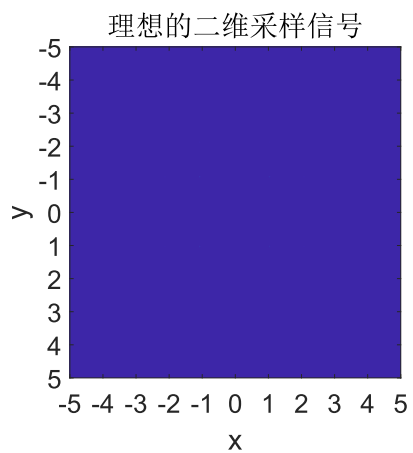
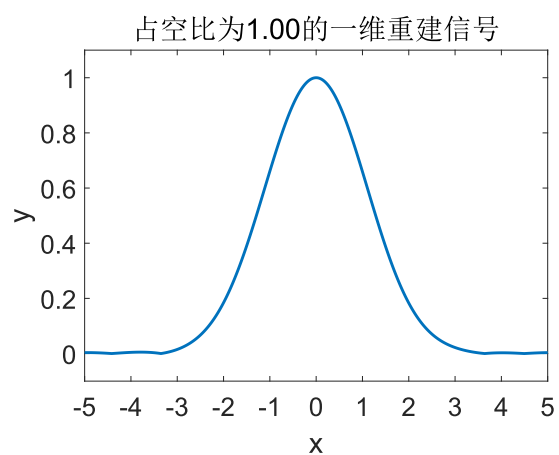
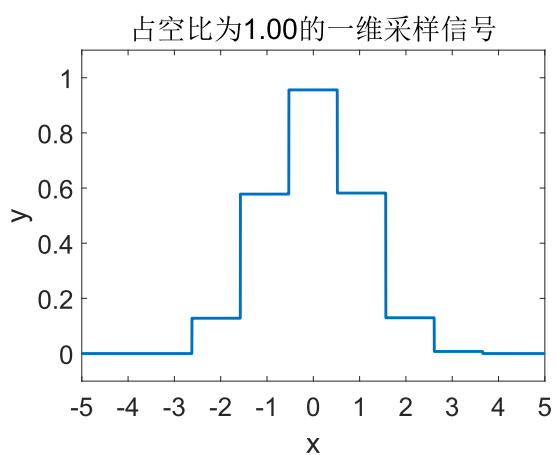
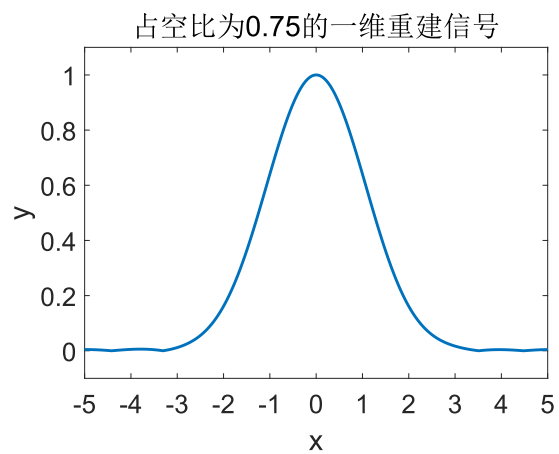
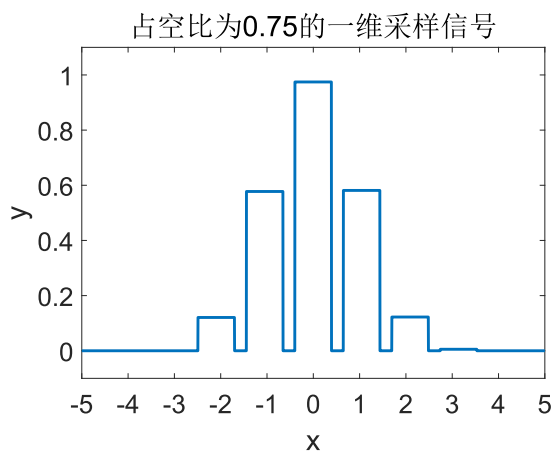


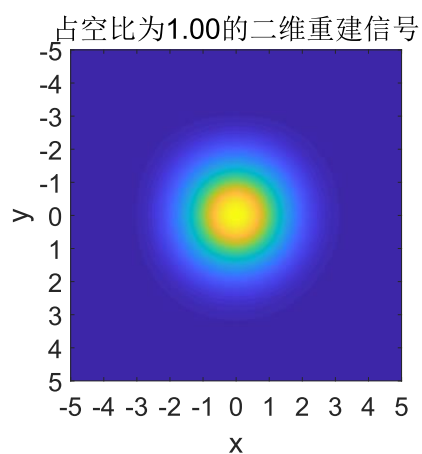
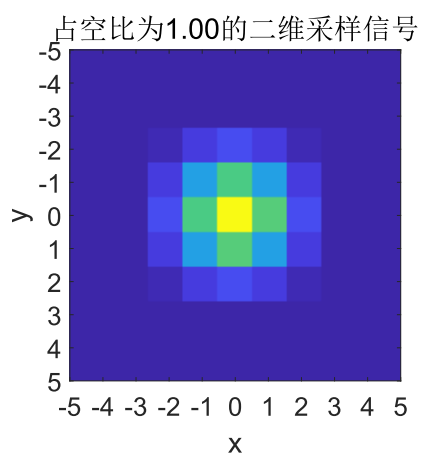
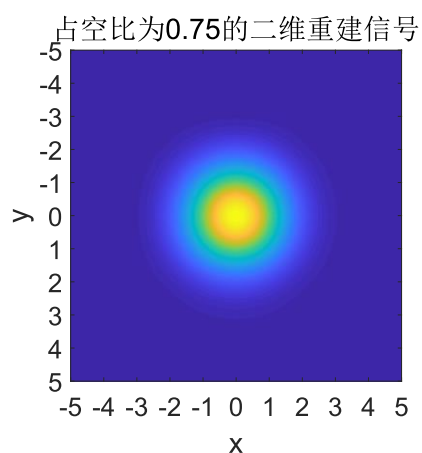
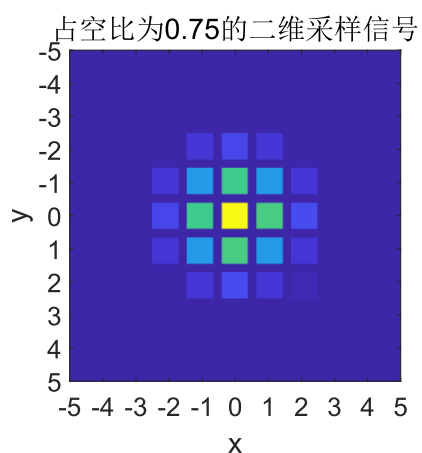
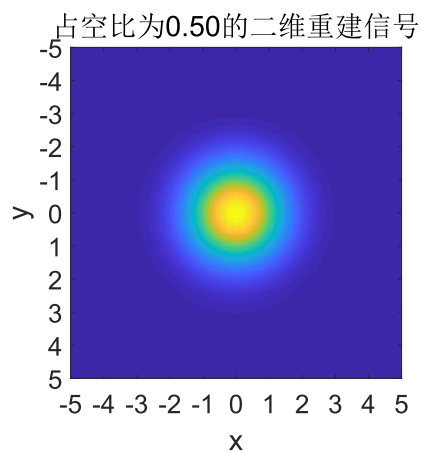
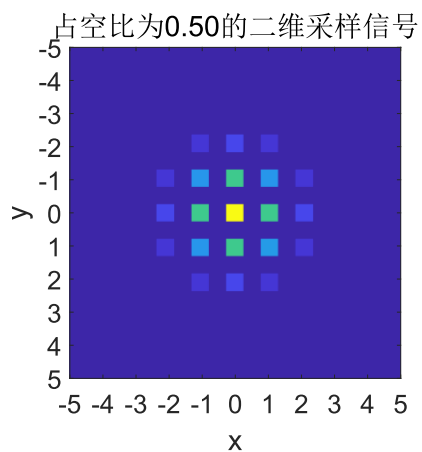


#### 4.3. 采样脉冲占空比对采样和重建的影响

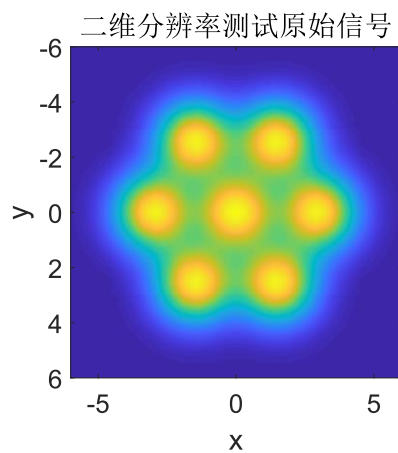
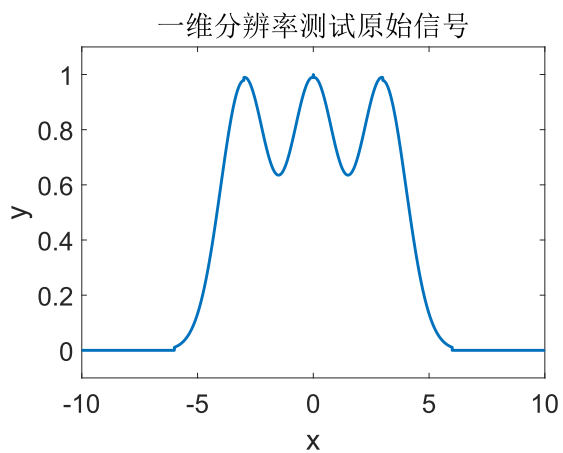


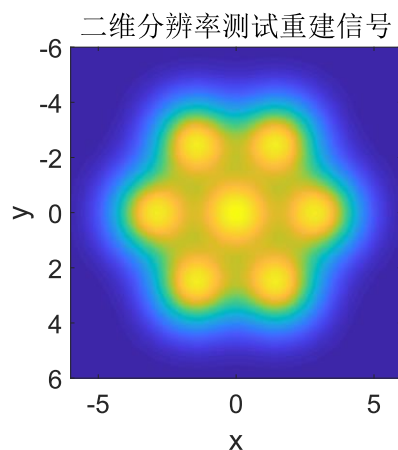
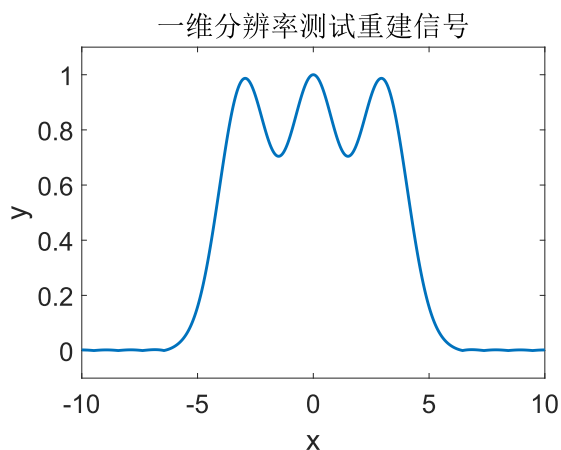
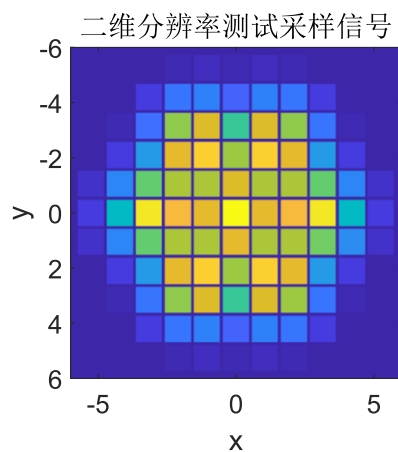
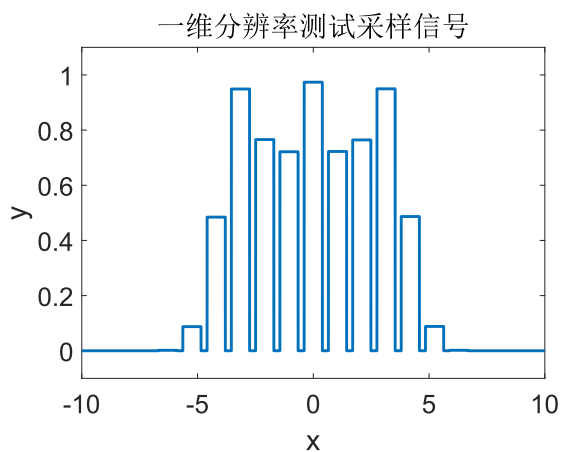






#### 4.4. 分辨率测试





## 5. 实验分析

### 5.1. 采样周期、采样率和信号重建

根据实验原理中的分析，对于标准差为 $\sigma$ 的高斯信号，在满足以下条件时进行采样，可以获得较好的重建效果。

$$x, y \in [-3\sigma, 3\sigma]$$

$$\omega_s \geq \frac{6}{\sigma} \quad T_s \leq \frac{\pi}{3} \sigma$$

对于一个二维平面上的高斯函数，有 25 个像素参与了采样。对这一采样信号分别进行频域滤波重建和空域sinc函数内插重建，在距离中心三倍标准差的范围内，两种方法都可以获得较好的重建效果。但是由于在进行空域内插时，sinc函数的范围，因此在离中心较远时，重建的误差较大。

### 5.2. 非理想采样和占空比

根据理论分析，矩形脉冲采样中脉冲序列的占空比并不会影响频域理想低通滤波重建的结果。

根据实验模拟，从理想采样（冲激串）到脉冲占空比为 1 的非理想矩形脉冲采样，频域低通滤波所获得的重建信号都没有区别。因此，矩形采样脉冲序列的占空比并不会影响采样信号的重建。

### 5.3. 空间分辨率

根据实验仿真结果，根据按照要求进行采样得到的信号中进行重建，得到的重建信号中可以有效分辨空间距离为三倍标准差的两个高斯函数脉冲。因此，这样的采样和重建方法可以用于成像系统的信号采集。

## 5.4. 成像系统的像素尺寸

由上述计算和仿真，我们可以得到以下结论。

对于面阵感光元件，我们假设其每个像素都是正方形，在空间中等间距排列形成像素阵列。

如果一个光学成像系统的点扩展函数可以近似拟合为标准差为 $\sigma$ 的高斯函数。则对其成像结果进行采集时所需要的面阵感光元件，相邻两个像素的中心的距离应不超过 $\frac{\pi}{3}\sigma \approx 1.047\sigma$ 。单个像素的尺寸不会影响对采样结果的重建，可以结合通光量等因素综合考虑具体取值。

在光学系统中，也常使用半高全宽（FWHM）来描述成像系统的分辨能力，其被定义为点扩展函数为最大值一半时的宽度。对于标准差为 $\sigma$ 的高斯函数，其半高全宽为 $FWHM = 2\sqrt{2\ln 2}\sigma \approx 2.355\sigma$ 。

因此，相邻两个像素的中心的距离满足：

$$\Delta x \leq \frac{\pi}{3} \cdot \sigma = \frac{\pi}{3} \cdot \frac{1}{2\sqrt{2\ln 2}} \cdot FWHM \approx 1.047 \cdot \sigma \approx 0.445 \cdot FWHM$$

## 6. 实验心得

本次实验的主要内容是信号的采样和重建。结合理论课上学习的关于卷积、傅里叶变换和空域采样的知识，我在实验中分别测试了使用冲激序列的理想采样和使用矩形脉冲序列的非理想采样。和之前的卷积与逆卷积实验不同，本次实验更接近于对实际的物理过程进行建模与 MATLAB 程序仿真的过程。因此在进行实验仿真之前需要构建一个合理的数学模型，以此指导实验中对于采样参数（如空间采样率和采样脉冲占空比）的设置。同时需要在一定程度上结合实际实现过程中元件的一些典型特征来进行仿真。

最后的仿真结果和理论计算的结果较为吻合，两种采样方法和相对应的重建方法都获得了不错的结果。同时我也进行了更进一步的空间分辨率仿真测试，来验证对单个衍射斑进行采样和重建的方法能否用于采样和重建多个衍射斑组合而成的信号。

但实验过程中仍然有一些需要解决的问题。首先，重建信号和原始信号会存在一定的误差，虽然可以通过 MATLAB 中提供的 plot、imagesc 和 mesh 等绘图函数直观地展示信号并半定量地分析重建质量，但是在实验中仍然缺少一个可以定量且不需要人工介入的重建质量分析方法。其次，对非理想采样信号在进行重建时，采样序列可以选取矩形脉冲序列或是相对应的冲激序列。这两种方式的重建结果优劣与实际实现难易仍需要讨论与验证。最后，仿真实验中并没有考虑噪声干扰和模拟-数字转换过程中引入的量化误差等在实际实现过程中不可避免的问题。