

DIP-HW1

黃川懿 520030910268

2022 年 10 月 3 日

摘要

本次实验的目的是熟悉掌握基于高斯函数的卷积模板及其图像平滑特点

目录

1 生成高斯函数的曲面显示	1
1.1 高斯曲面 ($\sigma = 5$)	1
1.2 总结	1
1.3 Matlab 源码	2
2 从高斯连续函数中计算五个不同大小的卷积模板	3
2.1 高斯函数标准差 σ 和高斯模板窗口大小关系	3
2.1.1 3σ 原则	3
2.1.2 离散贝塞尔近似 (与教学 ppt 较为一致)	3
2.1.3 OpenCV 源码	3
2.1.4 杨辉三角	4
2.2 确定 σ 后模板的计算	4
2.3 5*5 卷积模板	5
2.4 7*7 卷积模板	5
2.5 11*11 卷积模板	6
2.6 15*15 卷积模板	6
2.7 19*19 卷积模板	7
2.8 总结	7
2.9 Matlab 源码	7
3 利用 5 个不同大小的卷积模板进行平滑卷积	8
3.1 gray 高斯噪声 3.jpg	8
3.2 gray 高斯多尺度平滑 2.jpg	10
3.3 gray 高斯多尺度平滑 6.jpg	11
3.4 总结	11
3.5 Matlab 源码	12
4 利用 Laplacian 卷积模板增强各个多尺度平滑图像	13
4.1 gray 高斯噪声 3.jpg	13
4.2 gray 高斯多尺度平滑 2.jpg	15
4.3 gray 高斯多尺度平滑 6.jpg	16
4.4 总结	16
4.5 Matlab 源码	17
5 计算同一图像的多尺度平滑图像中每个像素的梯度	17
5.1 梯度的计算过程	17
5.2 gray 高斯噪声 3.jpg	18
5.3 gray 高斯多尺度平滑 2.jpg	20

5.4	gray 高斯多尺度平滑 6.jpg	21
5.5	总结	21
5.6	Matlab 源码	22
6	总结多尺度平滑图像中，噪声抑制的效果以及带来的负面影响	22

1 生成高斯函数的曲面显示

1.1 高斯曲面 ($\sigma = 5$)

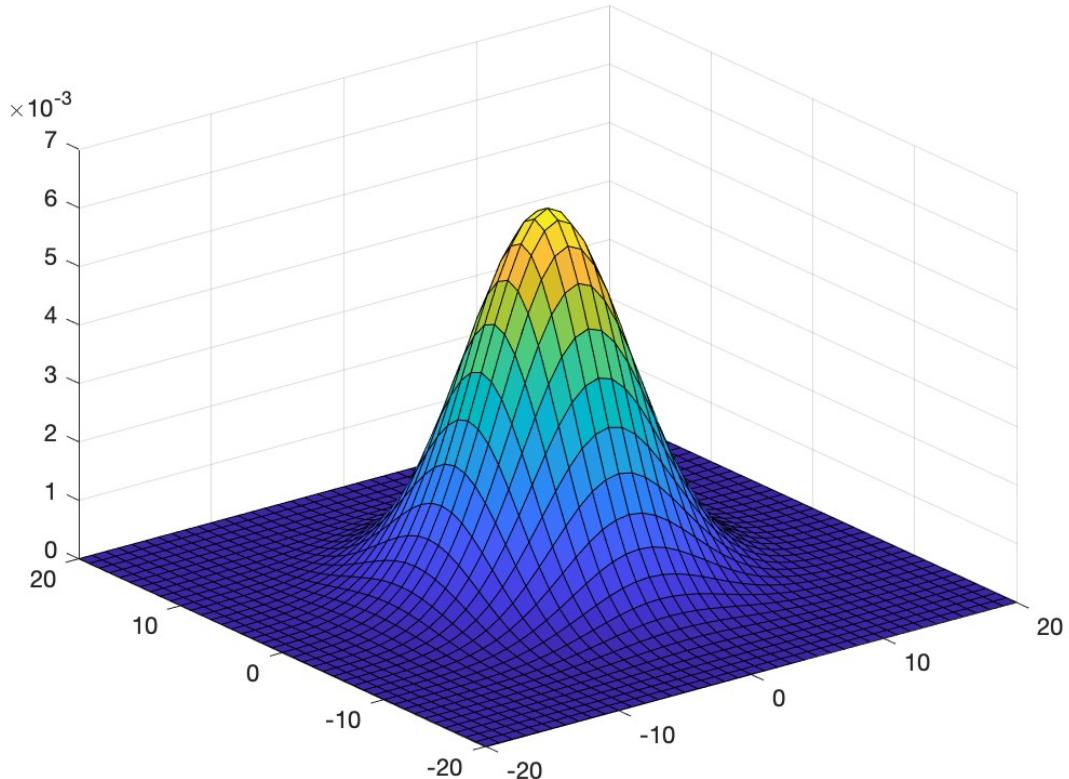


图 1: $\sigma = 5$ 时高斯曲面

此外，随文件附带展示曲面随 σ 变化的 gif 图像

1.2 总结

在图像中，已知 x, y 的相关性 $\mu=0$ (即中心为原点)，则待定参数只有 σ 1. 高斯曲面静态特点：

- (1) 曲面的等高线是从中心开始呈正态分布的同心圆
 - (2) 曲面呈现中心高且陡峭，四周低而缓的形态，其中大部分高度较高的点集中在 3 个 σ 以内，3 个 σ 以外基本高度为 0
 - (3) 具有旋转对称性，在各个方向上的平滑程度相同
2. 高斯曲面随 σ 变化的动态特点
- 从生成的 gif 可以看出：
- (1) 随着 σ 增大，中心的高度下降，中心向各方向的下降趋势减缓，图像的“陡峭”程度下降。

1.3 Matlab 源码

```
1 clear ;
2 clc ;
3
4 size=41;
5 start_==-(size -1)/2;
6 end_=(size -1)/2;
7 [X,Y] = meshgrid(start_:end_,start_:end_);
%生成\sigma=5时的高斯曲面
8 sigma_5=5;
9 name="Gauss_Surface.png";
10 h = exp(-(X.*X+Y.*Y)./(2*sigma_5^2))/(2*pi*sigma_5^2);
% 归一化
11 sumh = sum(h(:));
12 h= h/sumh;
13 surf(X,Y,h);
14 drawnow
%生成\sigma从1变化到5时高斯曲面变化的gif
15 filename="Gauss_Surface.gif";
16 fig=figure;
17 for sigma = 1:0.2:5
18     h = exp(-(X.*X+Y.*Y)./(2*sigma^2))/(2*pi*sigma^2);
% 归一化
19     sumh = sum(h(:));
20     h= h/sumh;
21     surf(X,Y,h);
22     xlabel('x');
23     ylabel('y');
24     zlabel('z');
25     axis([-20 20 -20 20 0 0.25]);
26     drawnow
27     frame=getframe(fig);
28     im=frame2im(frame);
29     [imind,cm]=rgb2ind(im,256);
30     if sigma == 1
31         imwrite(imind,cm,filename,'gif','Loopcount',inf)
32     end
33 end
```

```

36     else
37         imwrite(imind, cm, filename, 'gif', 'WriteMode', '
38             append);
39     end
end

```

2 从高斯连续函数中计算五个不同大小的卷积模板

2.1 高斯函数标准差 σ 和高斯模板窗口大小关系

高斯滤波使用的高斯核为 x 和 y 两个一维高斯的乘积，两个维度上的标准差 σ 通常相同，形式如下

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

当 $\mu = 0$ (相关性) 时，唯一需要控制的参数就是标准差 σ .

高斯核是对连续高斯函数的离散近似，通常对高斯曲面进行离散采样和归一化得出高斯核可以看成是与中心距离负相关的权重。平滑时，调整 σ 实际是在调整周围像素对当前像素的影响程度，调大 σ 即提高了远处像素对中心像素的影响程度，滤波结果也就越平滑。

在进行搜索后，得到以下几种 σ 和窗口大小的关系：

2.1.1 3σ 原则

高斯函数是钟形曲线，钟型曲线在区间 $(\mu-\sigma, \mu + \sigma)$ 范围内的面积占曲线下总面积的 68%。 $(\mu-2\sigma, \mu + 2\sigma)$ 范围占 95%， $(\mu-3\sigma, \mu + 3\sigma)$ 范围占 99.7%，一般 3σ 外的数值已接近于 0，可忽略，半径为 3σ 即窗口大小为 $6\sigma \times 6\sigma$ 即可，通常取最近的奇数

2.1.2 离散贝塞尔近似 (与教学 ppt 较为一致)

在 Mathematica 函数高斯矩阵特点计算高斯离散矩阵的几种方法，例如使用离散贝塞尔近似。默认情况下，半径 = $2 * \text{sigma}$ ，这意味着当 $\text{sigma} = 1$ 时，矩阵将是 5×5 。(取最近的奇数)

2.1.3 OpenCV 源码

```

809:     ... // automatic detection of kernel size from sigma
810:     if( ksize.width <= 0 && sigma1 > 0 )
811:         ksize.width = cvRound(sigma1*(depth == CV_8U ? 3 : 4)*2 + 1) | 1;
812:     if( ksize.height <= 0 && sigma2 > 0 )
813:         ksize.height = cvRound(sigma2*(depth == CV_8U ? 3 : 4)*2 + 1) | 1;
...

```

图 2: opencv 计算 kernelsize 源码

可以看出 $kernel size \sigma * 2 * 3 + 1 \quad \sigma * 2 * 4 + 1 \quad \sigma * 2 * 3 + 1 \quad 3\sigma$ 的原则一致 $\sigma * 2 * 3 + 1$ 一致

2.1.4 杨辉三角

帕斯卡三角形的行很好地近似于高斯函数，并且具有具有整数值的实际优势，其和为 2 的幂（我们可以将这些值精确地存储为整数、定点值或浮点数）。例如，假设要构造一个 7×7 的高斯核，我们可以使用 Pascal 三角形的第 7 行，如下所示：

$$\begin{array}{c}
 \begin{array}{ccccccc}
 & & 1 & & & & \\
 & & 1 & 1 & & & \\
 & & 1 & 2 & 1 & & \\
 & & 1 & 3 & 3 & 1 & \\
 & & 1 & 4 & 6 & 4 & 1 \\
 & & 1 & 5 & 10 & 10 & 5 & 1 \\
 & & 1 & 6 & 15 & 20 & 15 & 6 & 1
 \end{array} &
 \begin{array}{c}
 1/64
 \end{array} &
 \begin{array}{c}
 \left[\begin{array}{c}
 1 \\
 6 \\
 15 \\
 20 \\
 15 \\
 6 \\
 1
 \end{array} \right] *
 \left[\begin{array}{c}
 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 1 & 6 & 15 & 20 & 15 & 6 & 1
 \end{array} \right] 1/64
 \end{array} \\
 \left[\begin{array}{c}
 1 & 6 & 15 & 20 & 15 & 6 & 1 \\
 6 & 36 & 90 & 120 & 90 & 36 & 6 \\
 15 & 90 & 225 & 300 & 225 & 90 & 15 \\
 20 & 120 & 300 & 400 & 300 & 120 & 20 \\
 15 & 90 & 225 & 300 & 225 & 90 & 15 \\
 6 & 36 & 90 & 120 & 90 & 36 & 6 \\
 1 & 6 & 15 & 20 & 15 & 6 & 1
 \end{array} \right] 1/4096
 \end{array}$$

图 3: 杨辉三角计算高斯模板

我们可以利用 $i=j=k+1$ 处的元素来计算对应的 σ

$$\sigma = \frac{2^{2k}}{C_{2k} k \sqrt{2\pi}} \quad (2)$$

2.2 确定 σ 后模板的计算

高斯模板是通过高斯函数计算出来的。一个二维的高斯函数如下：

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

其中 (x, y) 为点坐标，在图像处理中可认为是整数； σ 是标准差。要想得到一个高斯滤波器的模板，可以对高斯函数进行离散化，得到的高斯函数值作为模板的系数。对于窗口模板

的大小为 $(2k + 1) \times (2k + 1)$, 模板中各个元素值的计算公式如下:

$$H_{i,j} = \frac{1}{2\pi\sigma^2} \exp^{-\frac{i^2+j^2}{2\sigma^2}} \quad (4)$$

计算出模板中各个元素值后, 要进行归一化:

$$H'_{i,j} = \frac{H_{i,j}}{\sum_{i=-k}^k \sum_{j=-k}^k H_{i,j}} \quad (5)$$

前文讨论中发现离散贝塞尔近似得到的结果与教学 ppt 给出的结果较为一致, 故取 $\sigma = (\text{size} - 1)/4$ 以上就是不同大小高斯卷积模板的计算公式。借助 Matlab 我们可以分别得出不同大小的高斯卷积模板如下:

2.3 5*5 卷积模板

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030

2.4 7*7 卷积模板

0.0013	0.0041	0.0079	0.0099	0.0079	0.0041	0.0013
0.0041	0.0124	0.0241	0.0301	0.0241	0.0124	0.0041
0.0079	0.0241	0.0470	0.0587	0.0470	0.0241	0.0079
0.0099	0.0301	0.0587	0.0733	0.0587	0.0301	0.0099
0.0079	0.0241	0.0470	0.0587	0.0470	0.0241	0.0079
0.0041	0.0124	0.0241	0.0301	0.0241	0.0124	0.0041
0.0013	0.0041	0.0079	0.0099	0.0079	0.0041	0.0013

2.5 11*11 卷积模板

0.0005	0.0010	0.0018	0.0026	0.0034	0.0036	0.0034	0.0026	0.0018	0.0010	0.0005
0.0010	0.0021	0.0036	0.0054	0.0069	0.0075	0.0069	0.0054	0.0036	0.0021	0.0010
0.0018	0.0036	0.0064	0.0095	0.0121	0.0131	0.0121	0.0095	0.0064	0.0036	0.0018
0.0026	0.0054	0.0095	0.0142	0.0180	0.0195	0.0180	0.0142	0.0095	0.0054	0.0026
0.0034	0.0069	0.0121	0.0180	0.0229	0.0248	0.0229	0.0180	0.0121	0.0069	0.0034
0.0036	0.0075	0.0131	0.0195	0.0248	0.0269	0.0248	0.0195	0.0131	0.0075	0.0036
0.0034	0.0069	0.0121	0.0180	0.0229	0.0248	0.0229	0.0180	0.0121	0.0069	0.0034
0.0026	0.0054	0.0095	0.0142	0.0180	0.0195	0.0180	0.0142	0.0095	0.0054	0.0026
0.0018	0.0036	0.0064	0.0095	0.0121	0.0131	0.0121	0.0095	0.0064	0.0036	0.0018
0.0010	0.0021	0.0036	0.0054	0.0069	0.0075	0.0069	0.0054	0.0036	0.0021	0.0010
0.0005	0.0010	0.0018	0.0026	0.0034	0.0036	0.0034	0.0026	0.0018	0.0010	0.0005

2.6 15*15 卷积模板

0.0003	0.0004	0.0007	0.0010	0.0013	0.0016	0.0018	0.0019	0.0018	0.0016	0.0013	0.0010	0.0007	0.0004	0.0003
0.0004	0.0007	0.0011	0.0017	0.0022	0.0027	0.0031	0.0032	0.0031	0.0027	0.0022	0.0017	0.0011	0.0007	0.0004
0.0007	0.0011	0.0018	0.0026	0.0035	0.0042	0.0048	0.0050	0.0048	0.0042	0.0035	0.0026	0.0018	0.0011	0.0007
0.0010	0.0017	0.0026	0.0038	0.0050	0.0061	0.0069	0.0072	0.0069	0.0061	0.0050	0.0038	0.0026	0.0017	0.0010
0.0013	0.0022	0.0035	0.0050	0.0066	0.0081	0.0092	0.0096	0.0092	0.0081	0.0066	0.0050	0.0035	0.0022	0.0013
0.0016	0.0027	0.0042	0.0061	0.0081	0.0100	0.0113	0.0118	0.0113	0.0100	0.0081	0.0061	0.0042	0.0027	0.0016
0.0018	0.0031	0.0048	0.0069	0.0092	0.0113	0.0128	0.0133	0.0128	0.0113	0.0092	0.0069	0.0048	0.0031	0.0018
0.0019	0.0032	0.0050	0.0072	0.0096	0.0118	0.0133	0.0139	0.0133	0.0118	0.0096	0.0072	0.0050	0.0032	0.0019
0.0018	0.0031	0.0048	0.0069	0.0092	0.0113	0.0128	0.0133	0.0128	0.0113	0.0092	0.0069	0.0048	0.0031	0.0018
0.0016	0.0027	0.0042	0.0061	0.0081	0.0100	0.0113	0.0118	0.0113	0.0100	0.0081	0.0061	0.0042	0.0027	0.0016
0.0013	0.0022	0.0035	0.0050	0.0066	0.0081	0.0092	0.0096	0.0092	0.0081	0.0066	0.0050	0.0035	0.0022	0.0013
0.0010	0.0017	0.0026	0.0038	0.0050	0.0061	0.0069	0.0072	0.0069	0.0061	0.0050	0.0038	0.0026	0.0017	0.0010
0.0007	0.0011	0.0018	0.0026	0.0035	0.0042	0.0048	0.0050	0.0048	0.0042	0.0035	0.0026	0.0018	0.0011	0.0007
0.0004	0.0007	0.0011	0.0017	0.0022	0.0027	0.0031	0.0032	0.0031	0.0027	0.0022	0.0017	0.0011	0.0007	0.0004
0.0003	0.0004	0.0007	0.0010	0.0013	0.0016	0.0018	0.0019	0.0018	0.0016	0.0013	0.0010	0.0007	0.0004	0.0003

2.7 19*19 卷积模板

0.0002	0.0002	0.0003	0.0005	0.0006	0.0008	0.0009	0.0010	0.0011	0.0011	0.0011	0.0010	0.0009	0.0008	0.0006	0.0005	0.0003	0.0002	0.0002
0.0002	0.0004	0.0005	0.0007	0.0009	0.0012	0.0014	0.0016	0.0017	0.0017	0.0017	0.0016	0.0014	0.0012	0.0009	0.0007	0.0005	0.0004	0.0002
0.0003	0.0005	0.0007	0.0010	0.0014	0.0017	0.0020	0.0023	0.0025	0.0025	0.0025	0.0023	0.0020	0.0017	0.0014	0.0010	0.0007	0.0005	0.0003
0.0005	0.0007	0.0010	0.0014	0.0019	0.0023	0.0028	0.0031	0.0034	0.0035	0.0034	0.0031	0.0028	0.0023	0.0019	0.0014	0.0010	0.0007	0.0005
0.0006	0.0009	0.0014	0.0019	0.0025	0.0031	0.0036	0.0041	0.0044	0.0045	0.0044	0.0041	0.0036	0.0031	0.0025	0.0019	0.0014	0.0009	0.0006
0.0008	0.0012	0.0017	0.0023	0.0031	0.0038	0.0045	0.0051	0.0055	0.0057	0.0055	0.0051	0.0045	0.0038	0.0031	0.0023	0.0017	0.0012	0.0008
0.0009	0.0014	0.0020	0.0028	0.0036	0.0045	0.0054	0.0061	0.0066	0.0067	0.0066	0.0061	0.0054	0.0045	0.0036	0.0028	0.0020	0.0014	0.0009
0.0010	0.0016	0.0023	0.0031	0.0041	0.0051	0.0061	0.0069	0.0075	0.0076	0.0075	0.0069	0.0061	0.0051	0.0041	0.0031	0.0023	0.0016	0.0010
0.0011	0.0017	0.0025	0.0034	0.0044	0.0055	0.0066	0.0075	0.0080	0.0082	0.0080	0.0075	0.0066	0.0055	0.0044	0.0034	0.0025	0.0017	0.0011
0.0011	0.0017	0.0025	0.0035	0.0045	0.0057	0.0067	0.0076	0.0082	0.0084	0.0082	0.0076	0.0067	0.0057	0.0045	0.0035	0.0025	0.0017	0.0011
0.0011	0.0017	0.0025	0.0034	0.0044	0.0055	0.0066	0.0075	0.0080	0.0082	0.0080	0.0075	0.0066	0.0055	0.0044	0.0034	0.0025	0.0017	0.0011
0.0010	0.0016	0.0023	0.0031	0.0041	0.0051	0.0061	0.0069	0.0075	0.0076	0.0075	0.0069	0.0061	0.0051	0.0041	0.0031	0.0023	0.0016	0.0010
0.0009	0.0014	0.0020	0.0028	0.0036	0.0045	0.0054	0.0061	0.0066	0.0067	0.0066	0.0061	0.0054	0.0045	0.0036	0.0028	0.0020	0.0014	0.0009
0.0008	0.0012	0.0017	0.0023	0.0031	0.0038	0.0045	0.0051	0.0055	0.0057	0.0055	0.0051	0.0045	0.0038	0.0031	0.0023	0.0017	0.0012	0.0008
0.0006	0.0009	0.0014	0.0019	0.0025	0.0031	0.0036	0.0041	0.0044	0.0045	0.0044	0.0041	0.0036	0.0031	0.0025	0.0019	0.0014	0.0009	0.0006
0.0005	0.0007	0.0010	0.0014	0.0019	0.0023	0.0028	0.0031	0.0034	0.0035	0.0034	0.0031	0.0028	0.0023	0.0019	0.0014	0.0010	0.0007	0.0005
0.0003	0.0005	0.0007	0.0010	0.0014	0.0017	0.0020	0.0023	0.0025	0.0025	0.0025	0.0023	0.0020	0.0017	0.0014	0.0010	0.0007	0.0005	0.0003
0.0002	0.0004	0.0005	0.0007	0.0009	0.0012	0.0014	0.0016	0.0017	0.0017	0.0017	0.0016	0.0014	0.0012	0.0009	0.0007	0.0005	0.0004	0.0002
0.0002	0.0002	0.0003	0.0005	0.0006	0.0008	0.0009	0.0010	0.0011	0.0011	0.0011	0.0010	0.0009	0.0008	0.0006	0.0005	0.0003	0.0002	0.0002

2.8 总结

可以发现高斯模板的系数具有以下特点：

- (1) 因为归一化处理过，所以高斯模板所有的系数之和为 1
- (2) 具备旋转对称性，即以原点中心对称，到原点有相同 D4 距离 (Manhattan distance) 的点的值相同
- (3) 原点（中心点）的系数最大，且随 D4 距离的增大而减小。
- (4) 模板 size 越大， σ 越大，系数随 D4 衰减的速率越小

2.9 Matlab 源码

```
1 clear;
2 clc;
3 % size 的值根据生成模板的大小变化
4 % 5*5 卷积核的 size 为 5
5 % 7*7 卷积核的 size 为 7
6 % 11*11 卷积核的 size 为 11
7 % 15*15 卷积核的 size 为 15
8 % 19*19 卷积核的 size 为 19
9 % 此处以 11*11卷积核为例
10 size=11;
11 start_ = -(size - 1) / 2;
12 end_ = (size - 1) / 2;
13 [X,Y] = meshgrid(start_:end_, start_:end_);
```

```
14 sigma = ( size -1) /4;
15 h = exp(-(X.*X+Y.*Y)./(2*sigma^2))/(2*pi*sigma^2);
16 % 归一化
17 sumh = sum(h(:));
18 h= h/sumh;
19 % surf(X,Y,h)
```

3 利用 5 个不同大小的卷积模板进行平滑卷积

3.1 gray 高斯噪声 3.jpg



(a) 5Kernel



(b) 7Kernel



(c) 11Kernel



(d) 15Kernel



(e) 19Kernel

3.2 gray 高斯多尺度平滑 2.jpg



(f) 5Kernel



(g) 7Kernel



(h) 11Kernel



(i) 15Kernel



(j) 19Kernel

3.3 gray 高斯多尺度平滑 6.jpg



(k) 5Kernel



(l) 7Kernel



(m) 11Kernel



(n) 15Kernel



(o) 19Kernel

3.4 总结

(1) 对于任一经过高斯模板处理滤波后的图像，可以看出图像变得更为平滑（即与周围点的灰度差距减小），同时一些细节变得模糊，甚至丢失了。这是由于高斯滤波器是低通滤波器，滤掉了一些图像的高频部分，剩下低频部分，(2) 对于存在噪声的图像，进过滤波后，噪声被一定程度地抑制 (3) 可以看出，卷积模板越大，滤波后的图像就平滑，丢失的细节也会越多，整体看起来就越模糊。这是因为模板越大，那么一个像素的信息就会被均分到更多的邻域里去，因此整体看起来就会更平滑，而细节也就越不凸显

3.5 Matlab 源码

```
1 clear ;
2 clc ;
3 path =char(strcat(cd,filesep));
4 images = dir([path,'*.jpg']);
%resluts 用于记录 GaussBlur时卷积的结果
5 results=cell(3,5);
% GaussBlur
6
7
8 folder='./gaussblur_result/';
9
10
11 if ~exist(folder,'dir')
12     mkdir(folder);
13 end
14 kernel_size=[5,7,11,15,19];
15 for image_index=1:length(images)
16     for kernel_index=1:5
17         size_=kernel_size(kernel_index);
18         sigma=(size_-1)/4;
19         start_-=(size_-1)/2;
20         end_=(size_-1)/2;
21         [X,Y]=meshgrid(start_:end_,start_:end_);
22         h=exp(-(X.*X+Y.*Y)./(2*sigma^2))/(2*pi*sigma^2);
23 % 归一化
24         sumh=sum(h(:));
25         h=h/sumh;
26 % 用生成的高斯模板和图像卷积
27         img=imread([path,images(image_index).name]);
28 %记录卷积结果
29         gauss_conved=conv2(h,img);
30         results{image_index,kernel_index}=gauss_conved;
31 %利用自带函数填充
32         new_img=imfilter(img,h,"replicate");
33         forename=strsplit(images(image_index).name,'.');
34         endname=sprintf("_%dKernel.jpg",size_);
35         filename=strcat('gaussblur_result',filesep,
36                         forename{1},endname);
```

```
36     imwrite(new_img,filename);  
37 end  
38 end
```

4 利用 Laplacian 卷积模板增强各个多尺度平滑图像

为了更直观地分析增强效果的异同，故报告展示仅做过 laplacian 处理的图像，而不是展现于原图叠加后的图像，增强后的图像在附件中

4.1 gray 高斯噪声 3.jpg



(p) 5Kernel



(q) 7Kernel



(r) 11Kernel

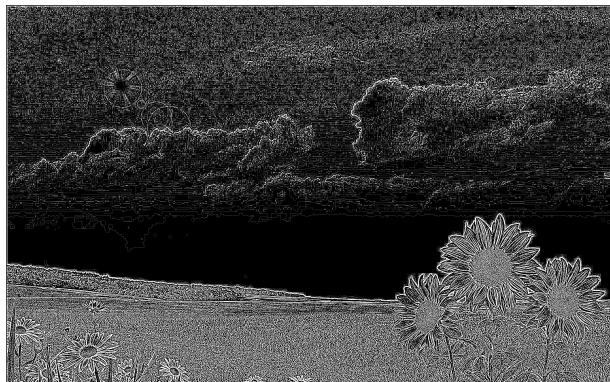


(s) 15Kernel



(t) 19Kernel

4.2 gray 高斯多尺度平滑 2.jpg



(u) 5Kernel



(v) 7Kernel



(w) 11Kernel



(x) 15Kernel



(y) 19Kernel

4.3 gray 高斯多尺度平滑 6.jpg



(z) 5Kernel



(a) 7Kernel



(b) 11Kernel



(c) 15Kernel



(d) 19Kernel

4.4 总结

Laplace 算子是一种各向同性算子，二阶微分算子。对于图像中孤立的点，边缘的响应强于其他地方，故

(1) 同：同一图像的多尺度平滑图像 laplacian 处理后的边缘处，孤立点和其他地方的对比度差异变大，产生了一定锐化的效果。(2) 异：随着同一图像的多尺度平滑图像的 kernel size 的增大，laplacian 处理后中对于孤立的点，边缘的产生的响应数量在不断减少，而增强的区域也在减少，增强效果相对减弱，直观感受就是图中的亮点数量在减少，边缘

在变细。这是因为高斯滤波的 kernel size 越大，图像的边缘与其相邻点的差异越小，就越难使得 laplace 算子产生响应。

4.5 Matlab 源码

```
1 %其中 resluts 为 GaussBlur 时卷积的结果
2 % Laplacian
3 folder= './ laplacian_result /';
4 laplacian_kernel=[0 -1 0; -1 4 -1;0 -1 0];
5 if ~exist( folder , ' dir ')
6     mkdir( folder );
7 end
8 for image_index =1:length( images )
9     for kernel_index=1:5
10         lap_conved=conv2( laplacian_kernel , results {
11             image_index , kernel_index });
12         forename=strsplit( images( image_index ). name , ' . ' );
13         filename = strcat( " laplacian_result " , filesep ,
14             forename( 1 ) , ' _ ' , ...
15             num2str( kernel_size( kernel_index )) , ' Kernel ' ,
16             '_laplacian.jpg " );
17         % 利用 im2uint8 归一化生成图像
18         imwrite( im2uint8( lap_conved ) , filename );
19     end
20 end
```

5 计算同一图像的多尺度平滑图像中每个像素的梯度

5.1 梯度的计算过程

由于图像是离散函数，在某点的梯度我们不用求导，而是通过向前差商、向后差商或者中心差商获得

首先需要用到计算 x、y 方向导数值的两个滤波核

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

之后分别与图像进行卷积

把得到的值写在一起得到图像梯度向量

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \quad (6)$$

得到梯度的方向

$$\theta = \tan^{-1}\left(\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}}\right) \quad (7)$$

得到梯度的幅度

$$||\nabla f|| = \sqrt{\frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}} \quad (8)$$

为了使得可视化，仅展示经过 S_x 处理过的图像

5.2 gray 高斯噪声 3.jpg



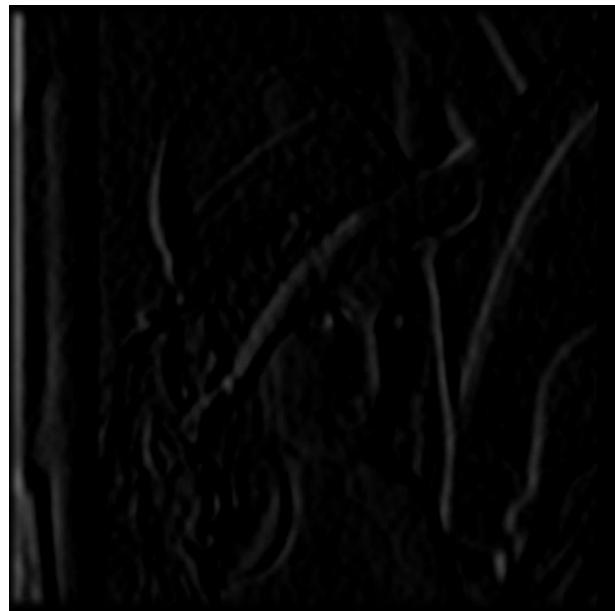
(e) 5Kernel



(f) 7Kernel



(g) 11Kernel



(h) 15Kernel



(i) 19Kernel

5.3 gray 高斯多尺度平滑 2.jpg



(j) 5Kernel



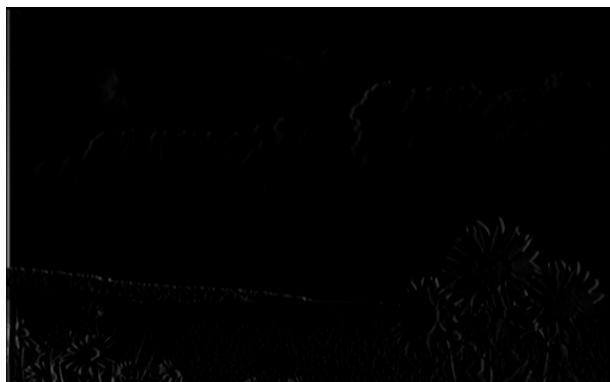
(k) 7Kernel



(l) 11Kernel



(m) 15Kernel



(n) 19Kernel

5.4 gray 高斯多尺度平滑 6.jpg



(o) 5Kernel



(p) 7Kernel



(q) 11Kernel



(r) 15Kernel



(s) 19Kernel

5.5 总结

通过观察各图像多尺度平滑计算梯度生成的图像，可以看出梯度算子对于该方向下的与相邻点的差距比较大的像素点的响应较大，比如该方向上的边缘点。而随着 Gauss 滤波的 kernel size 的增大，和梯度算子产生响应的区域越少，直观感受就是图中的亮点数量在减少，边缘在变细。这是因为高斯滤波的 kernel size 越大，图像的边缘与其相邻点的差异越小，就越难使得梯度算子产生响应。

5.6 Matlab 源码

```
1 % Gradient
2 %其中 resluts 为 GaussBlur时卷积的结果
3 folder= './ gradient_result /';
4 Prewitt_Kernel=[1,0,-1;1,0,-1;1,0,-1];
5 if ~exist ( folder , ' dir ')
6     mkdir ( folder );
7 end
8 for image_index =1:length ( images )
9     for kernel_index=1:5
10         gra_conved=conv2 ( Prewitt_Kernel , results {
11             image_index , kernel_index });
12         forename=strsplit ( images ( image_index ). name , ' . ' );
13         filename = strcat ( " gradient_result " , filesep ,
14             forename ( 1 ) , ' _ ' , ...
15             num2str ( kernel_size ( kernel_index )) , ' Kernel ' ,
16             '_gradient.jpg " );
17         % 利用 uint8 生成图像
18         imwrite ( uint8 ( gra_conved ) , filename );
19     end
20 end
```

6 总结多尺度平滑图像中，噪声抑制的效果以及带来的负面影响

噪声抑制的效果：

(1) 总体来看，高斯滤波能在不太大程度的情况下对噪声进行抑制。以 gray 高斯噪声 3 的多尺度平滑为例，可以看出原图中存在一些无关信息的白点，而在核大小为 5 时就可以明显感觉到白点被消除了很多，而随着 kernel size 的增大，消除的效果越好。这是因为高斯滤波在对图像邻域内像素进行平滑时，邻域内不同位置的像素被赋予不同的权高斯值，对图像进行平滑的同时能够更多的保留图像的总体灰度分布特征。

(2) 高斯滤波的 kernel size 越大，对于噪声的抑制效果越好，因为越多的相邻点的图像信息被分配到噪声对应区域中，噪声越趋向于相邻点的平均值，使得不再是与图像无关的信息。

负面影响：

尽管高斯滤波在对图像邻域内像素进行平滑时，邻域内不同位置的像素被赋予不同的

权高斯值，但是高斯滤波仍然会一定程度模糊化图像，以三张图片的多尺度平滑为例，在 kernel size 大于 11 时，原图好像被加上了一层蒙版，卷积核越大，看起来越模糊。可以看出，太大的卷积核会导致图像的失真。