

Python+OpenCV 图像处理

图像平滑

高斯滤波

讲师：李大羊

让临近的像素具有更高的重要度。对周围像素计算加权平均值，较近的像素具有较大的权重值。

23	158	140	115	131	87	131
238	0	67	16	247	14	220
199	197	25	106	156	159	173
94	149	40	107	5	71	171
210	163	198	226	223	156	159
107	222	37	68	193	157	110
255	42	72	250	41	75	184
77	150	17	248	197	147	150
218	235	106	128	65	197	202

×

0.05	0.1	0.05
0.1	0.4	0.1
0.05	0.1	0.05

=

			164			

$$(40 \times 0.05 + 107 \times 0.1 + 5 \times 0.05) + (198 \times 0.1 + 226 \times 0.4 + 223 \times 0.1) + (37 \times 0.05 + 68 \times 0.1 + 193 \times 0.05)$$

让临近的像素具有更高的重要度。对周围像素计算加权平均值，较近的像素具有较大的权重值。

23	158	140	115	131	87	131
238	0	67	16	247	14	220
199	197	25	106	156	159	173
94	149	40	107	5	71	171
210	163	198	226	223	156	159
107	222	37	68	193	157	110
255	42	72	250	41	75	184
77	150	17	248	197	147	150
218	235	106	128	65	197	202



1	1	2	1	1
1	3	4	3	1
2	4	8	4	2
1	3	4	3	1
1	1	2	1	1

11

A 10x10 grid of squares. The square at row 5, column 5 (using 1-based indexing from top-left) is colored red. All other squares are white.

GaussianBlur函数

```
dst = cv2.GaussianBlur( src , ksize , sigmaX )
```

GaussianBlur函数

```
dst = cv2.GaussianBlur( src , ksize , sigmaX )
```

src : 原始图像

ksize : 核大小

sigmaX: X方向方差

GaussianBlur函数

```
dst = cv2.GaussianBlur( src , ksize , sigmaX )
```

src : 原始图像，要处理的源图像

GaussianBlur函数

`dst = cv2.GaussianBlur(src , ksize , sigmaX)`

ksize : 核大小 (N,N) 必须是奇数

0.05	0.1	0.05
0.1	0.4	0.1
0.05	0.1	0.05

(3,3)

1	1	2	1	1
1	3	4	3	1
2	4	8	4	2
1	3	4	3	1
1	1	2	1	1

(5,5)

1	1	1	1	2	1	1
1	2	2	2	2	2	1
1	2	4	4	4	2	1
1	2	4	8	4	2	1
1	2	4	4	4	2	1
1	2	2	2	2	2	1
1	1	2	2	2	1	1

(7,7)

GaussianBlur函数

`dst = cv2.GaussianBlur(src , ksize , sigmaX)`

sigmaX: X方向方差，控制权重

1	1	2	1	1
1	2	4	2	1
2	4	8	4	2
1	4	4	2	1
1	1	2	1	1

0	0	1	0	0
0	1	2	1	0
1	2	3	2	1
0	1	2	1	0
0	0	1	0	0

0	3	6	3	0
1	4	7	4	1
2	5	8	5	2
1	4	7	4	1
0	3	6	3	0

GaussianBlur函数

`dst = cv2.GaussianBlur(src , ksize , sigmaX)`

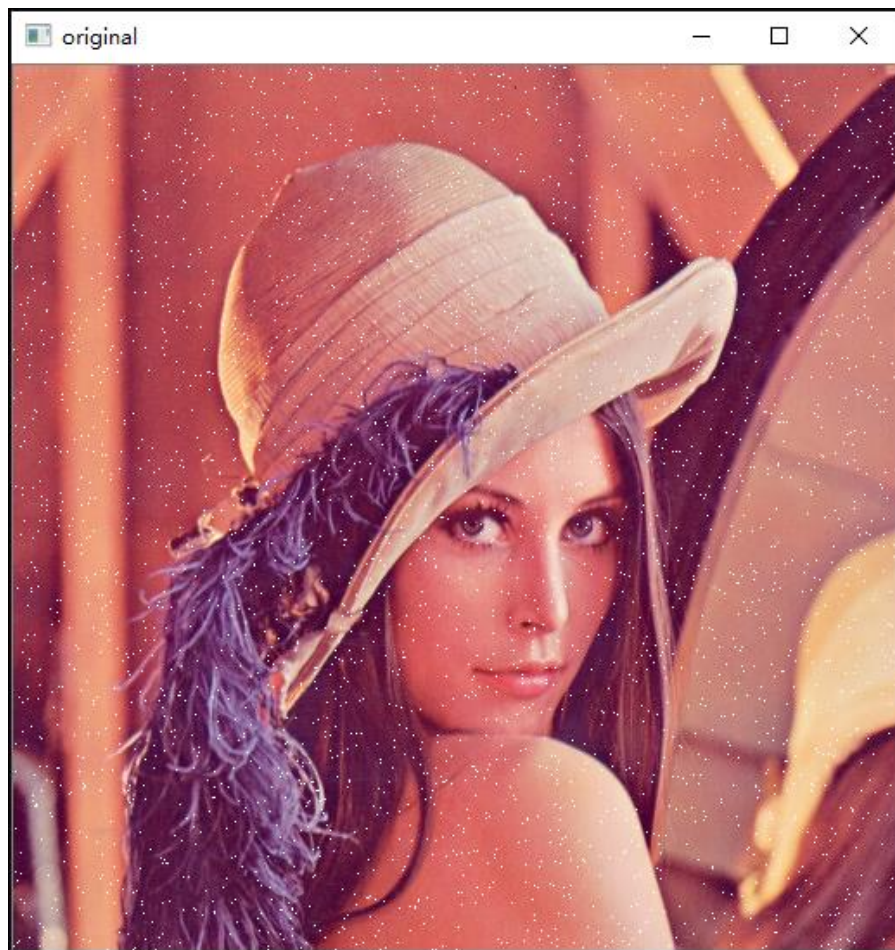
sigmaX: X方向方差，控制权重

sigmaX=0时: $\text{sigma} = 0.3 * ((\text{ksize} - 1) * 0.5 - 1) + 0.8$

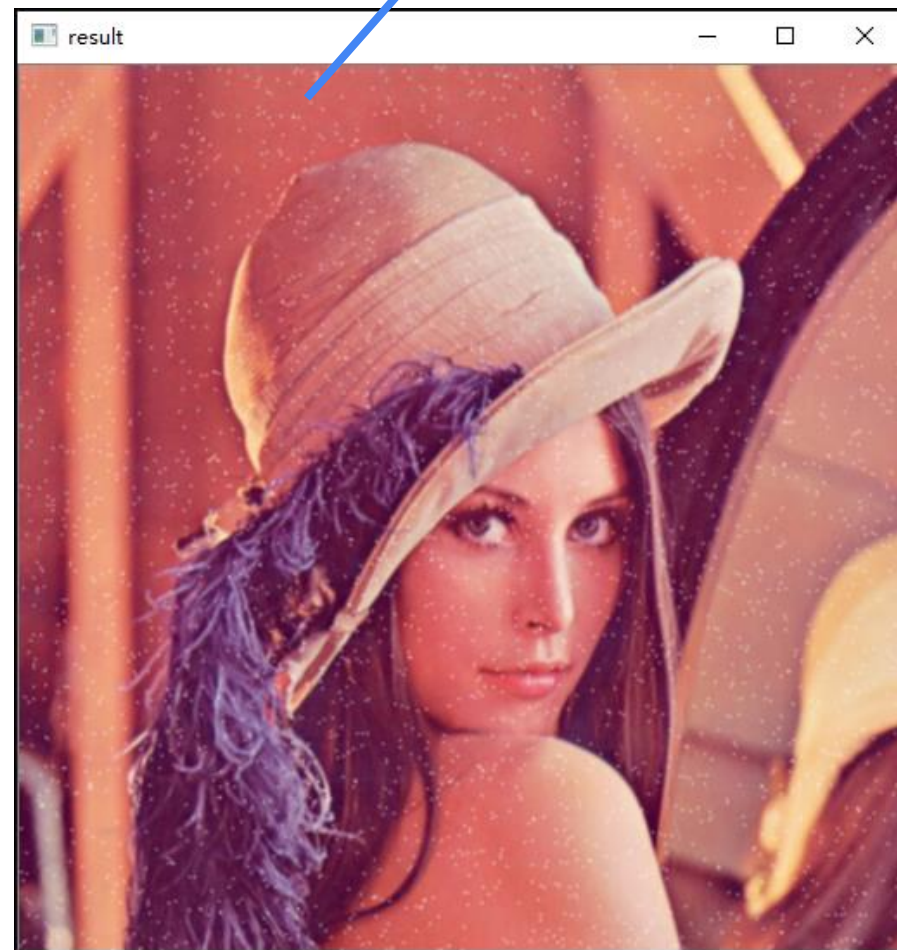
```
import cv2  
o=cv2.imread("image\\lenaNoise.png")  
r=cv2.GaussianBlur(o,(3,3),0)  
cv2.imshow("original",o)  
cv2.imshow("result",r)  
cv2.waitKey()  
cv2.destroyAllWindows()
```

高斯滤波

```
r=cv2.GaussianBlur(o,(3,3),0)
```



原始图像



目标图像

Python+OpenCV图像处理

图像平滑

高斯滤波

讲师：李大羊

lilizong@gmail.com