

Document Smoothing

I: Blur

1: Blur (Low-Pass Filtering)

LPF

Blur

- Kernel với mỗi phần tử có giá trị bằng 1, kích thước kernel do mình chỉ định sau đó tính trung bình các pixel trong kernel.
- Hình ảnh minh họa

$$K = \frac{1}{\text{ksize.width} * \text{ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ \dots & & & & & \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}$$

2: MedianBlur

MedianBlur

- Kernel với sẽ chuyển về mảng 1 chiều đã được sắp xếp pixel trung tâm sẽ được thay bằng trung vị của mảng.
- Ví Dụ: mảng đã được sắp xếp {3,4,5,6,7,7,8,9,10} thì median sẽ là 7.

3: GaussianBlur

Gaussian Smoothing

- Kernel sau khi tính toán theo hàm Gauss sẽ có dạng ma trận mà tâm của ma trận sẽ lớn nhất và nhỏ dần khi mở rộng ra ngoài biên.
- Hình ảnh minh họa

$$\frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

4: BilateralFilter

BilateralFilter

- Loại bỏ nhiễu nhưng vẫn giữ được biên. BilateralFilter cũng sử dụng bộ lọc Gauss nhưng có thêm một bước là nó chỉ xét những điểm có mức xám tương đồng với pixel trung tâm để làm mịn do những điểm ảnh ở biên có sự thay đổi mức xám rất rõ ràng.

5: Filter2D

- Convolution với kernel mình tạo ra

Tạo kernel

```
float array[3][3]={0, -0.25, 0}, {-0.25, 2, -0.25}, {0, -0.25, 0}};

Mat kernel(Kernel_size, Kernel_size, CV_32F, array);
```

- link [Filter2D](#)

6: BoxFilter

- Giống với Blur boxFilter cũng dùng Mat::ones/(ksizeWidth*KsizeHeight) nhưng boxfilter có thêm tham số độ sâu của ảnh và nếu normalize = False thì kernel sẽ là chỉ là Mat::ones.
- Link [BoxFilter](#)

II: Sharpen

1: High-Pass Filtering

- link [HPF](#)
- Sử dụng bộ lọc với nhân là dương và các phần tử xung quanh là âm.

0	-1	0
-1	4	-1
0	-1	0

-1	-2	-1
-2	12	-2
-1	-2	-1

.

2: