

Фильтрация изображений.
Свертка, 2D фильтр.
Шумоподавление.

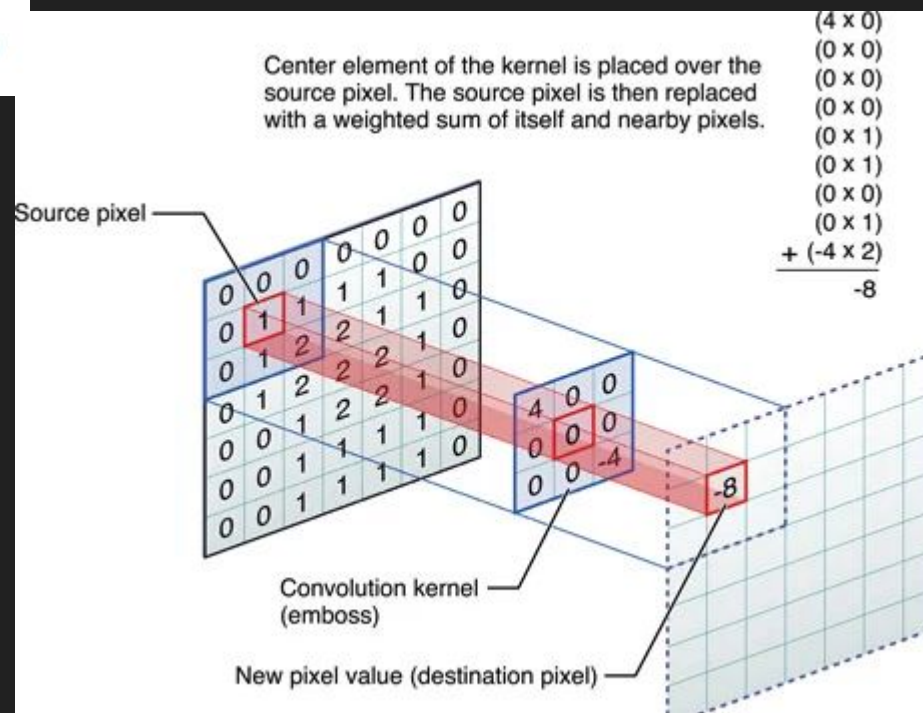
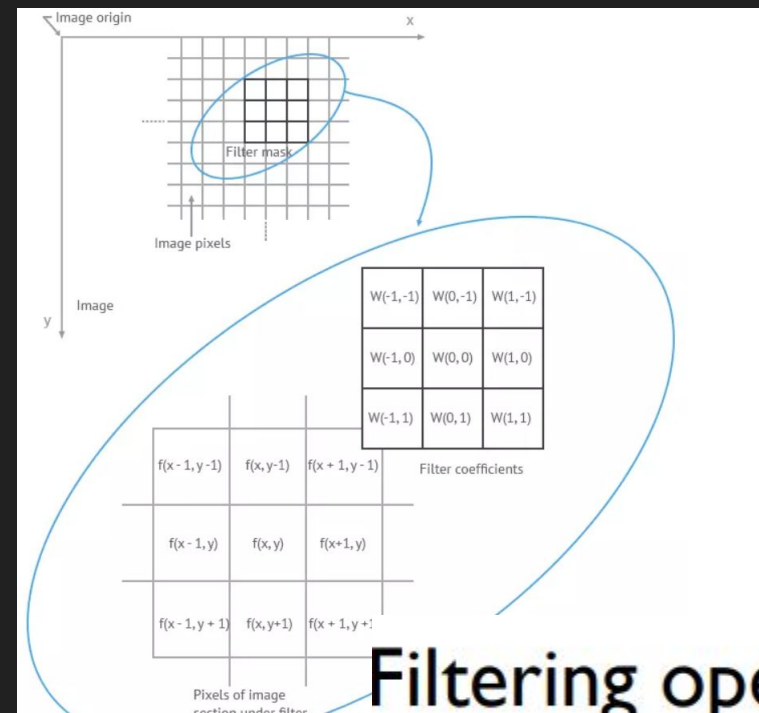
Цифровые кафедры, 2024

Свертка

Свертка

$$x(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2)$$

filter, mask, kernel, or template



Filtering operation called **cross-correlation**

Свертка

Коммутативность:

$$f * g = g * f.$$

Ассоциативность:

$$f * (g * h) = (f * g) * h.$$

Линейность (дистрибутивность по сложению и ассоциативность с умножением на скаляр):

$$(f_1 + f_2) * g = f_1 * g + f_2 * g,$$

$$f * (g_1 + g_2) = f * g_1 + f * g_2,$$

$$(af) * g = a(f * g) = f * (ag), \quad \forall a \in \mathbb{R}.$$

Правило дифференцирования:

$$D(f * g) = Df * g = f * Dg,$$

где Df обозначает производную функции f по любой переменной.

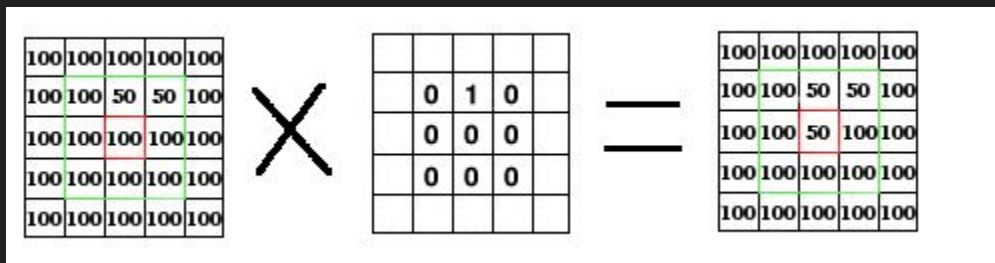
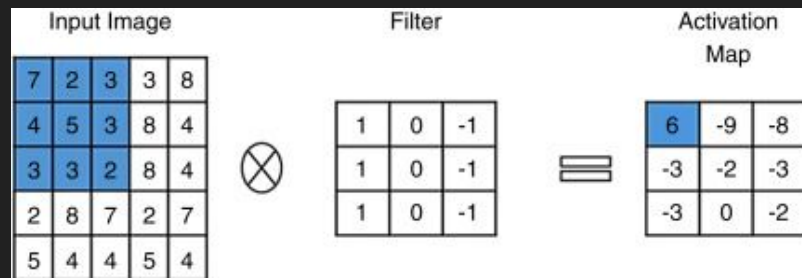
Преобразование Лапласа:

$$\mathcal{L}\{f(x) * g(x)\} = \mathcal{L}\{f(x)\} \cdot \mathcal{L}\{g(x)\}.$$

Свойство фурье-образа:

$$\mathfrak{F}[f * g] = \mathfrak{F}[f] \cdot \mathfrak{F}[g],$$

где $\mathfrak{F}[\]$ обозначает преобразование Фурье функции.



Несколько слов про нелинейную фильтрацию

Шумоподавление

Источники шума

- ошибки оборудования;
- плохие условия съемки (ночная съемка);
- помехи передачи сигнала по аналоговым каналам;
- дефекты утечки заряда (черный дефект).

Шум считывания

Каждый раз, когда вы считываете изображение с камеры, в изображение включается некоторый шум.

- ▶ **Даже если нет никакого сигнала** (никакого света, никакого заметного темнового тока), у вас все равно **будет шум в изображении**, оно всё равно не будет идеальным.
- ▶ Этот шум называют шумом считывания. Его генерирует и электроника устройства в целом, и непосредственно электроника сенсора.

Дробовой шум

Если съемка данных производится в течение секунды, может оказаться, что какой-то пиксель в сенсоре поражают 10 фотонов, а до другого доберутся 12, до третьего - 9.

- ▶ Такие отклонения всегда возникают на световом пути. Их и называют **дробовым шумом** (другое название – *фотонный шум*), который подчиняется распределению Пуассона.

Избежать дробового шума невозможно, при этом его величина растет как квадратный корень из интенсивности. Т.е. чем более яркий объект вы снимаете, тем больше получаете шума.

Темновой шум

- ▶ Закройте объектив камеры крышкой и сделайте несколько снимков с разной выдержкой. Вы заметите, что изображение тем ярче, чем больше выдержка. Это из-за темнового тока.
- ▶ *Темновой ток - малый электрический ток, который протекает через фоточувствительный детектор, например, фотодиод, фотоэлектронный умножитель, полупроводниковый детектор гамма-квантов и др. при отсутствии поглощенных фотонов.*
- ▶ Интенсивность удваивается, когда вы удваиваете длительность экспозиции, и, кроме того, почти удваивается каждые 6 градусов по шкале Цельсия.
- ▶ Некоторые пиксели светлеют быстрее других («горячие» пиксели светлеют очень быстро), в результате чего появляется узор стабильного пространственного шума.

Ошибка квантования

Когда мы считываем напряжение с сенсора, это аналоговый сигнал. Чтобы превратить его в цифровой, используется аналогово-цифровой преобразователь (АЦП).

- ▶ Предположим, у нас есть 8-битный АЦП.
- ▶ Установив его, мы можем записать 256 градаций интенсивности (2^8 в 8-й степени). Предположим далее, что наша ПЗС-матрица может накопить до насыщения 10 000 электронов.
- ▶ Если мы хотим использовать весь динамический диапазон ПЗС-матрицы, то должны установить АЦП так, чтобы каждое деление шкалы соответствовало примерно 25 фотонам ($10\,000/256$). Так, 10 на АЦП означает, что было захвачено примерно 250 фотонов, а 11 - около 275.

Не нужно много усилий, чтобы заметить здесь некоторую проблему. Для нас теперь нет разницы между 250 фотонами и 251, 255 или 260 фотонами. Им всем соответствует одно и то же значение. Эту проблему называют ошибкой квантования

Цветовые модели

- ▶ Чувствительность глаза к деталям цветовой информации (не к яркостной)
- ▶ Шум сильная зашумленность в цветовой компоненте (не в яркостной)
- ▶ Выход: работать в HSV, L^*a^*b (а не в RGB)
- ▶ Величина шума – от одного до нескольких пикселей

Метод главных компонент

- ▶ Основная идея – найти такой базис, в котором разброс значений будет минимален, проводить фильтрацию в этом базисе.

Шумы



световой шум



цветовой шум

Моделирование шума

$$f(i, j) = s(i, j) + n(i, j)$$

$f(i, j)$ - результирующее изображение

$s(i, j)$ - идеальное изображение сцены

$n(i, j)$ - шум

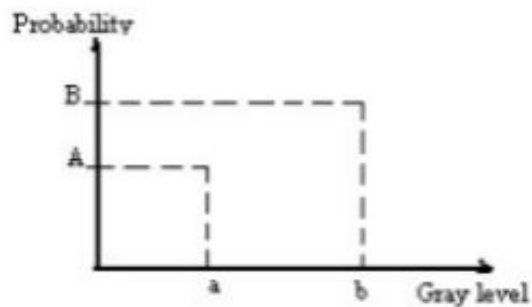
Генератор шума со случайным распределением

- ▶ Положения «шумовых пикселей» искать при помощи генератора псевдослучайных чисел (например, 5% от общего количества пикселей)
- ▶ Имеем генератор псевдослучайных чисел с равномерным законом распределения на интервале (0,1) (A)
- ▶ Надо получить новую случайную величину N с
$$\begin{bmatrix} n_1 n_2 \dots n_N \\ p_1 p_2 \dots p_N \end{bmatrix}$$
 распределением P
- ▶ интервал значений (0,1) разбивают на N интервалов, длины которых соответственно равны $\frac{p_1 p_2 \dots p_N}{a}$
- ▶ Получаем значение случайной величины
- ▶ Если это значение попадает в интервал i , наступило событие

Типы шума

- яркостный
 - интенсивность
- хроматический
 - цветовой

Шум “соль-перец”



Salt & pepper noise

Fig. 10.1 Probability density function for the salt & pepper noise model.

$$PDF_{salt \& \ pepper} = \begin{cases} A & \text{for } g = a \text{ ("pepper")} \\ B & \text{for } g = b \text{ ("salt")} \end{cases}$$



Шум Гаусса

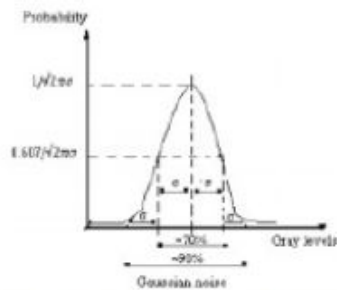


Fig. 10.2 Probability density function for the Gaussian noise model

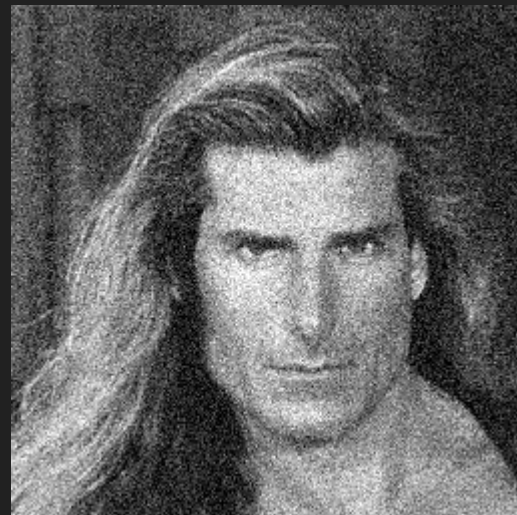
$$PDF_{Gaussian} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g-\mu)^2}{2\sigma^2}}$$

where:

g = gray level;

μ = mean;

σ = standard deviation;



Постоянный шум

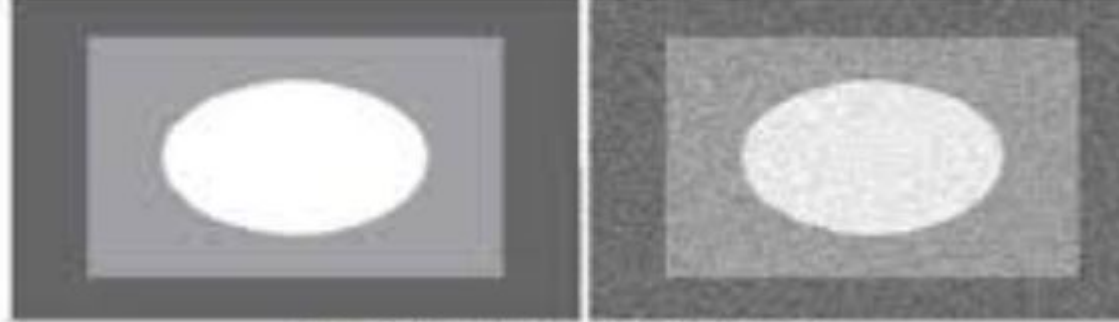
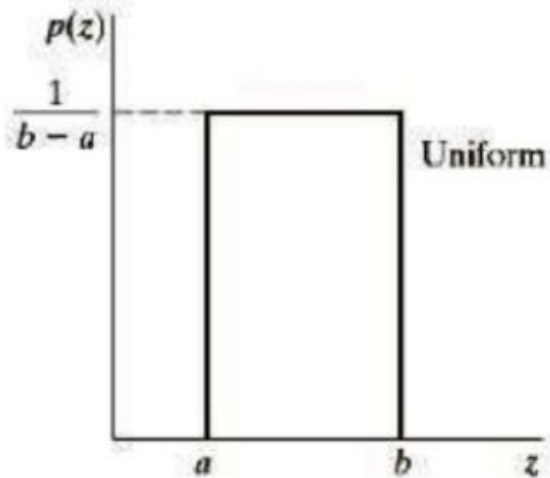


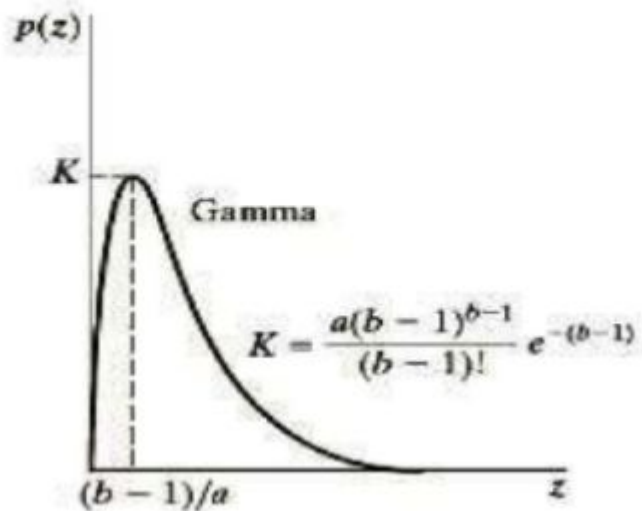
Fig. 5. Example of Uniform Noise.

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \quad [5]$$



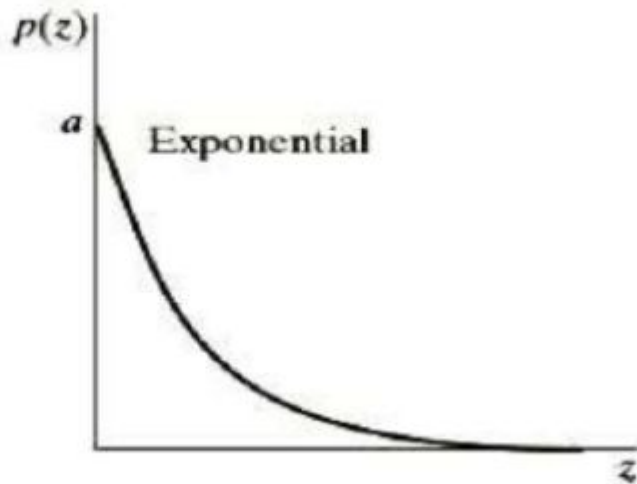
Гамма-шум, Шум Эрланга

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az}, & \text{for } z \geq 0 \\ 0, & \text{otherwise} \end{cases}$$



Экспоненциальный шум

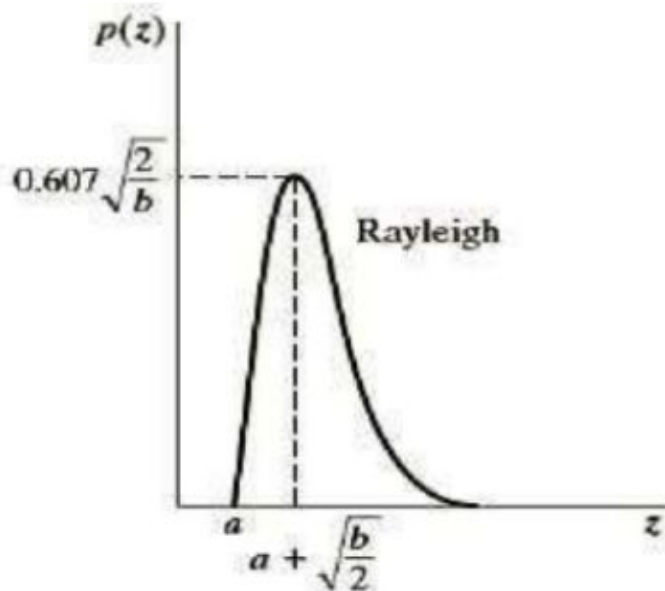
$$p(z) = \begin{cases} a e^{-az}, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad [5]$$



Гамма-шум при $b=1$

Шум Райли

$$p(z) = \begin{cases} \frac{2}{b} (z - a) e^{-(z-a)^2/b}, & \text{for } z \geq a \\ 0, & \text{for } z < a \end{cases} \quad [5]$$



Другие типы шумов

- ▶ Периодический шум
 - ▶ Генерируется во время «захвата» изображения
 - ▶ Параметры описываются при помощи преобразования Фурье
 - ▶ Удаление происходит при помощи анализа и изменения спектра Фурье
- ▶ Квантовый(фотонный) шум.
 - ▶ Возникает на рентгеновских снимках

Оценка распределения

- ▶ Для каждого сенсора рассчитать свою функцию распределения, сделав снимок поверхности с постоянным значением цветовой компоненты.
- ▶ Для периодического шума – оценка спектра Фурье.

Виды применяемых фильтров

- ▶ Точечные?
- ▶ Локальные – ДА!
- ▶ Глобальные?

- ▶ Основное понятие – свертка (но не всегда!!!).
- ▶ Например, упорядочивающие фильтры – не сверточные.

Виды фильтров

- сглаживающие (сверточные) фильтры;
- упорядочивающие фильтры;
- Фильтры на основе оценке распределений по интенсивности. дальности: гаусса, билатериальный, нелокальных средних
- фильтр с настраиваемой величиной коррекции;
- фильтрация в пространстве Фурье;
- фильтры вдоль доминантного направления (например, перпендикулярному направлению градиента);
- фильтры математической морфологии.

Сглаживающие фильтры

- ▶ Арифметическое среднее $\hat{f}(x, y) = \frac{1}{mn} \sum_{(s, t) \in S_{xy}} g(s, t)$

- ▶ Геометрическое среднее

- ▶ Меньше теряет деталей изображе

$$\hat{f}(x, y) = \left[\prod_{(s, t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

- ▶ Гармоническое среднее

- ▶ Гауссов и “белый” шум +
 - ▶ “Черный” шум -

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s, t) \in S_{xy}} g(s, t)}$$

- ▶ Контргармоническое средн

- ▶ $Q < 0$ - удаление “белого” шум
 - ▶ $Q > 0$ - удаление “черного” шум

$$\hat{f}(x, y) = \frac{\sum_{(s, t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s, t) \in S_{xy}} g(s, t)^Q}$$

Адаптивная фильтрация

- ▶ В фильтрации не участвуют пиксели, значение которых слишком отличается от значения центрального рассматриваемого пикселя.
- ▶ Или участвуют, но со специально подсчитанными весами.

Метод Гаусса

- ▶ Сепарабелен
- ▶ Фильтр низких частот
- ▶ Подходит для удаления постоянного и гауссовского шума
- ▶ Удаляет границы на изображении
- ▶ Как выбрать сигма? Зависит от задачи и что Вы знаете о ней
 - ▶ Например, 2% от размера диагонали изображения
- ▶ Важно лишь взаимное расположение

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$P \quad GB[I]_p = \sum_{q \in S} G_{\sigma}(\|p - q\|) I_q,$$

$$G_{\sigma}(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right).$$

Билатериальный фильтр (адапт)

- ▶ Не всегда лучший
- ▶ Часто очень хорош

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_p - I_q) I_q$$

$$W_p = \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(I_p - I_q)$$

- ▶ Важно и взаимное расположение пикселей и разница в цветовой компоненте
- ▶ Как найти оптимальные параметры?

Фильтр нелокальных средних

Non-local means (NL-means) –
веса зависят от близости целых
блоков, а не отдельных
пикселей

$v(x_{i,j})$ – блок вокруг
пикселя $x_{i,j}$

$$y_{i,j} = \sum_{k,m \in Q} x_{i+k,j+m} \cdot W(i,j,k,m)$$

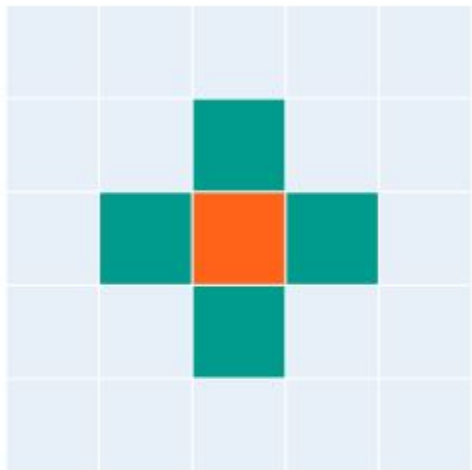
$$W(i,j,k,m) \approx \exp \left(- \frac{\|v(x_{i,j}) - v(x_{i+k,j+m})\|^2}{h^2} \right)$$

Упорядочивающие фильтры

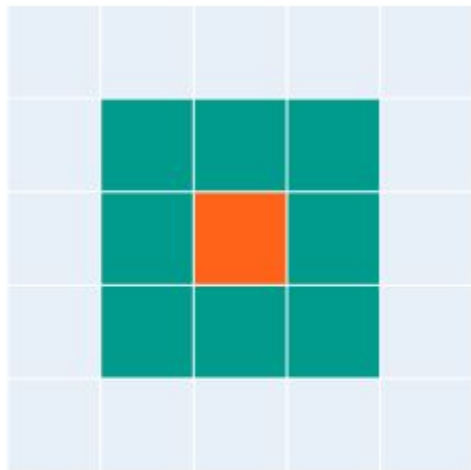
- ▶ Медианный фильтр $\hat{f}(x, y) = \underset{(s, t) \in S_{xy}}{\text{median}} \{g(s, t)\}$
 - ▶ Удаление шума “соль-перец”
- ▶ Фильтр максимума $\hat{f}(x, y) = \max_{(s, t) \in S_{xy}} \{g(s, t)\}$
 - ▶ Удаление “черного” шума
- ▶ Фильтр “минимума” $\hat{f}(x, y) = \min_{(s, t) \in S_{xy}} \{g(s, t)\}$
 - ▶ Удаление “белого” шума
- ▶ Фильтр “средней точки” $\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s, t) \in S_{xy}} \{g(s, t)\} + \min_{(s, t) \in S_{xy}} \{g(s, t)\} \right]$
 - ▶ Удаление Гауссова шума
 - ▶ Постоянного шума
- ▶ Усреднение с альфа-сдвиг $\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s, t) \in S_{xy}} g_r(s, t)$
 - ▶ Удалить $d/2$ наибольших и $d/2$ наименьших значений

Методы математической морфологии

4- и 8- связанное соседство



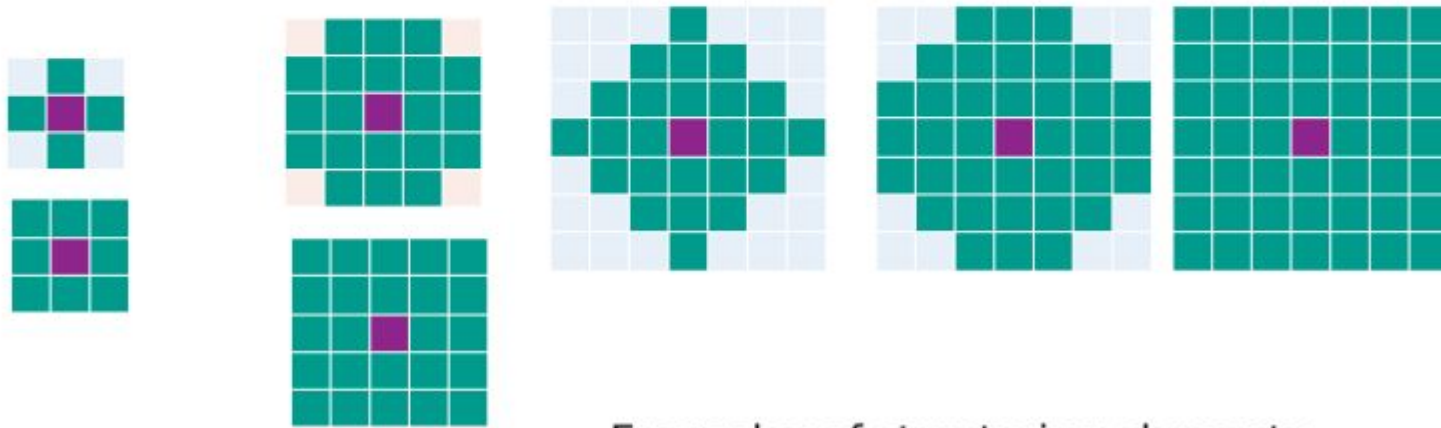
four-neighborhood
N4



eight-neighborhood
N8

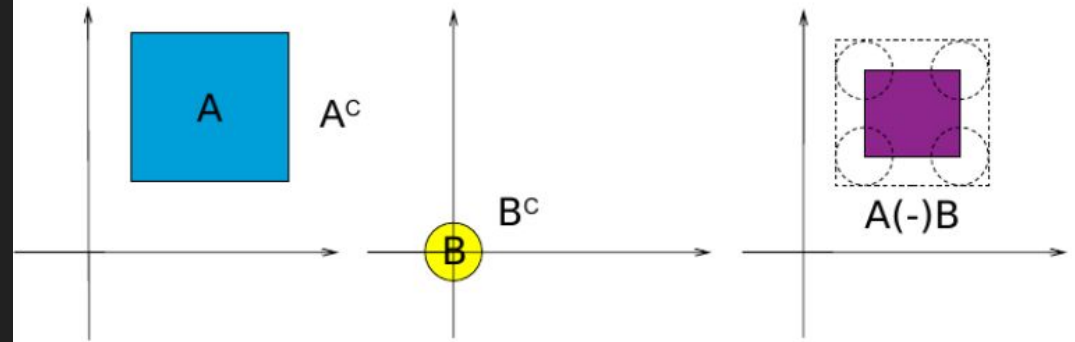
Структурный элемент

- ▶ A structuring element is simply a binary image (or mask) that allows us to define arbitrary neighborhood structures.

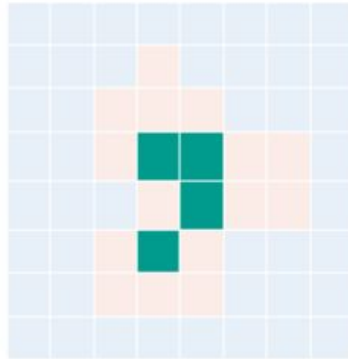
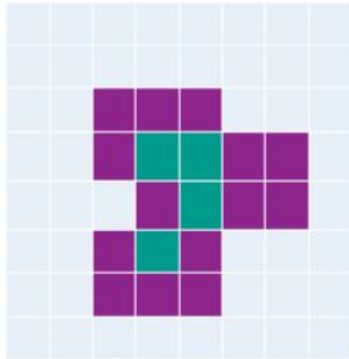
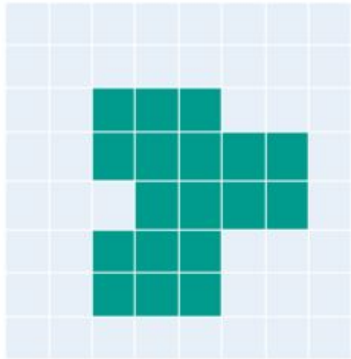


Examples of structuring elements

Ерозия, Сужение



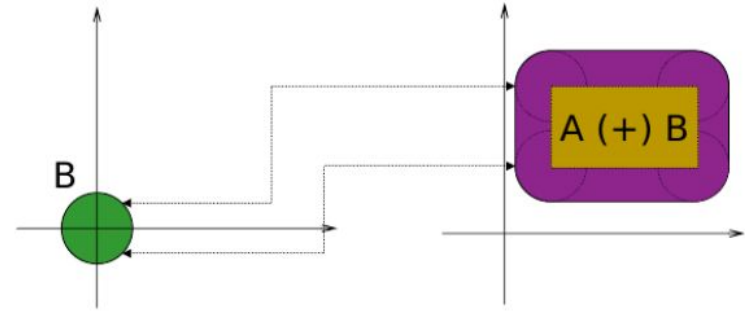
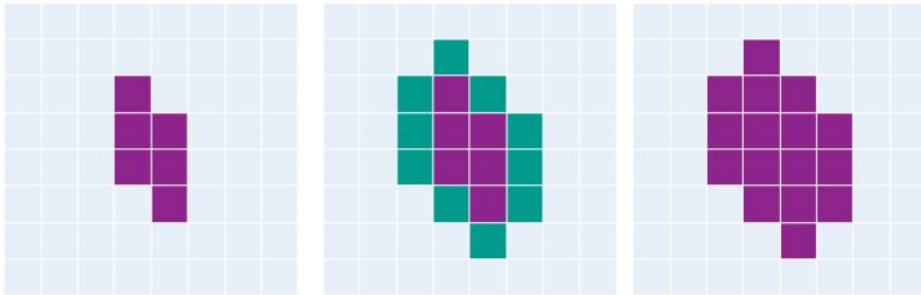
- Change a foreground pixel to background if it has a background pixel as a 4-neighbor.



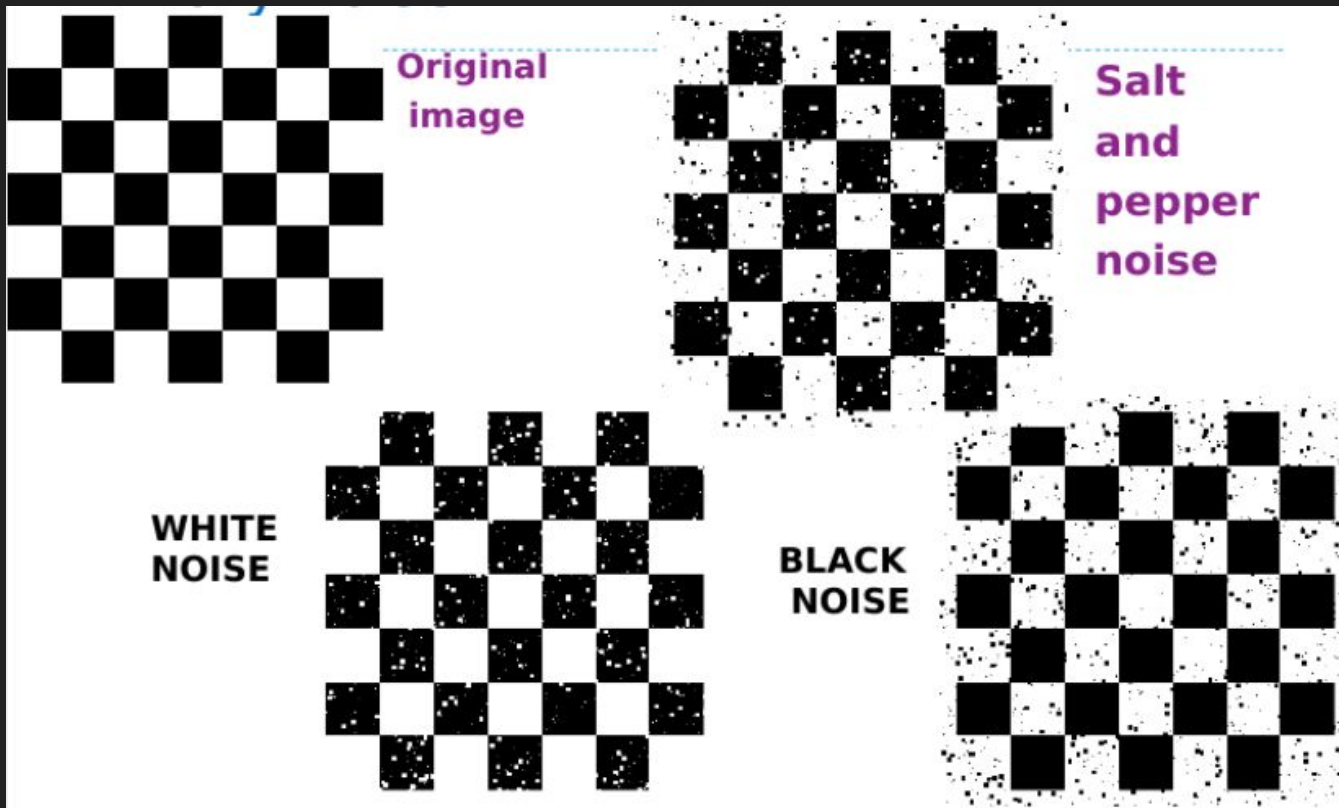
Дилатация, Расширение

Dilation Example

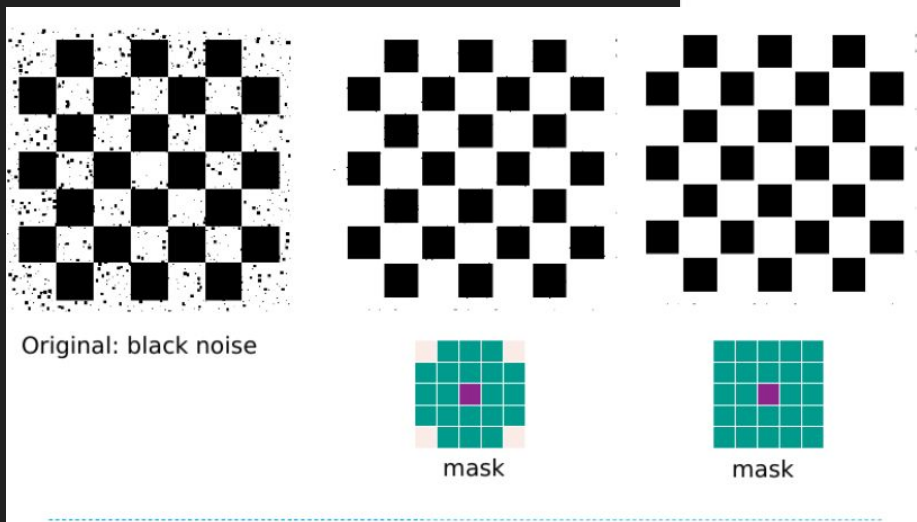
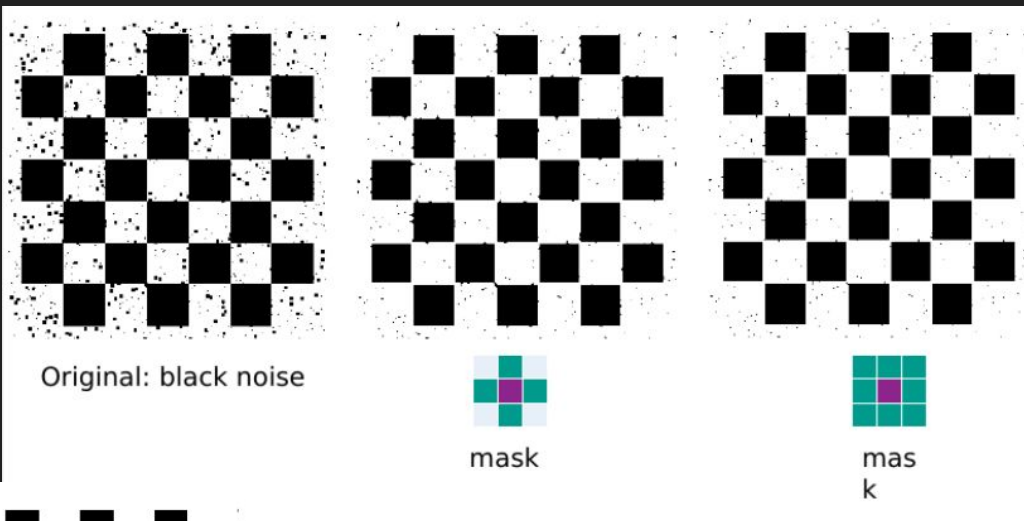
- Change a background pixel to foreground if it has a foreground pixel as a 4-neighbor.



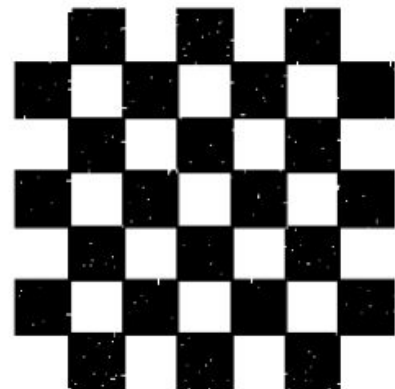
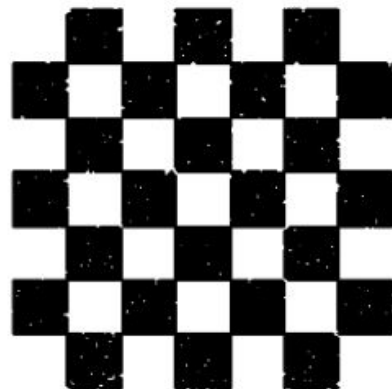
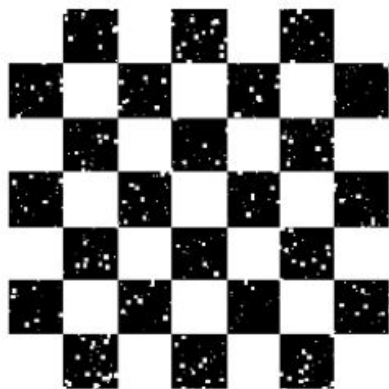
Удаление шума соль-перец



Черный шум



Белый шум



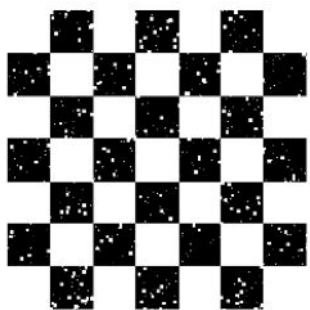
Original: white noise



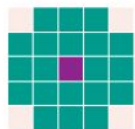
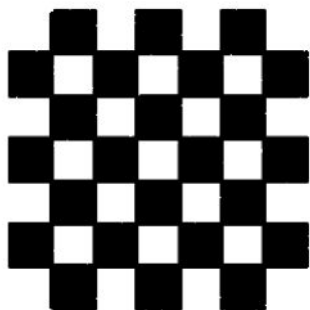
mask



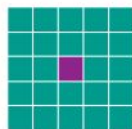
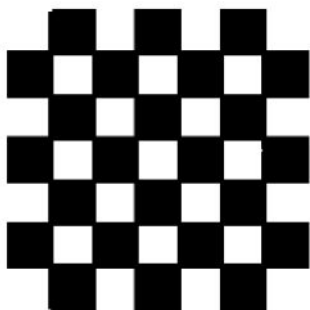
mask
k



Original: white noise



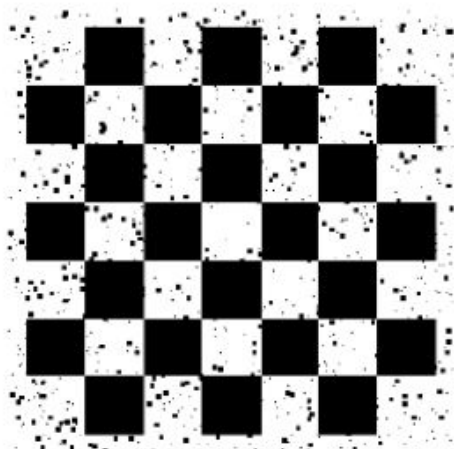
mask



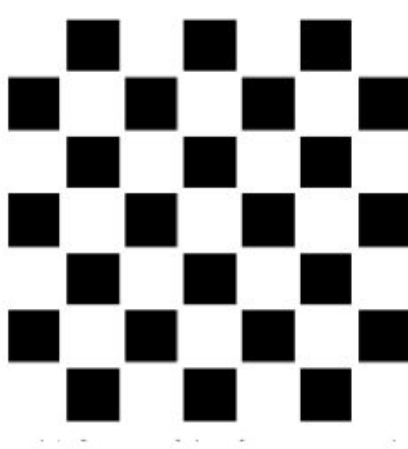
mask

Заметили проблему?

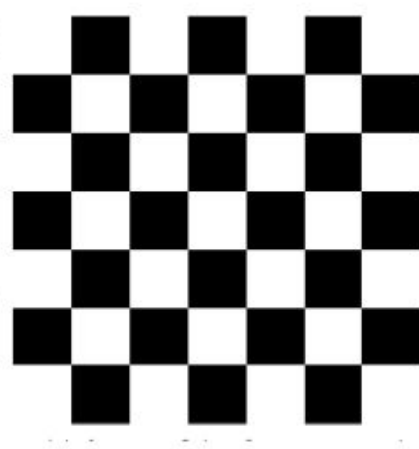
Closing results



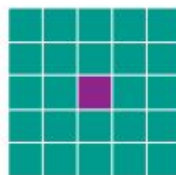
Original: black noise



Dilation

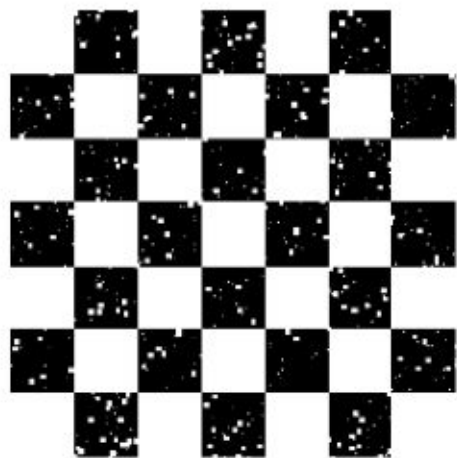


Closing

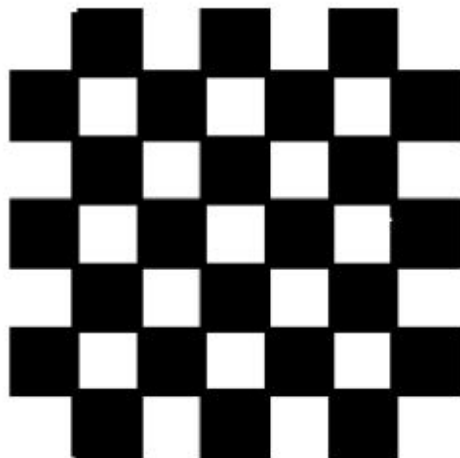


mask

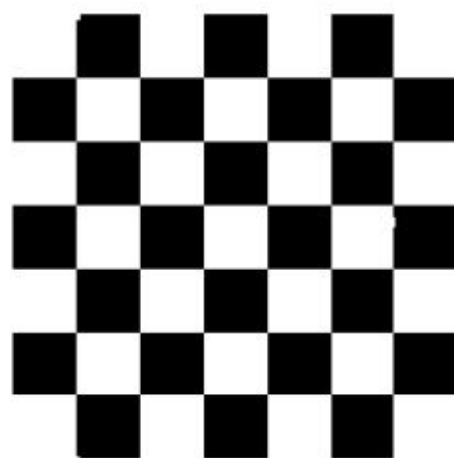
Opening results



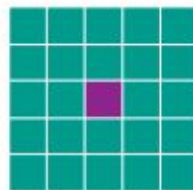
Original: white noise



Erosion



Opening



mask

Какой фильтр можно было бы применить для удаления шума типа соль-перец для бинарных изображений?