




Фильтрация изображений

- *Под фильтрацией изображений понимают операцию, имеющую своим результатом изображение того же размера, полученное из исходного по некоторым правилам (фильтрам). Обычно цвет каждого пикселя результирующего изображения обусловлена цветами пикселей, расположенных в некоторой его окрестности в исходном изображении.*
 - *Фильтрация изображений необходима при реализации фундаментальных операций компьютерного зрения, распознавания образов и обработки изображений.*
- 



Линейные фильтры

- Пусть задано исходное полутоновое изображение A , и обозначим интенсивности его пикселей $A(x, y)$. **Линейный фильтр** определяется вещественнозначной функцией F , заданной на растре. Данная функция называется **ядром фильтра**, а сама фильтрация производится при помощи операции дискретной свертки (взвешенного суммирования):

$$B(x, y) = \sum_i \sum_j F(i, j) \cdot A(x + i, y + j).$$

- Результатом служит изображение B . Обычно ядро фильтра отлично от нуля только в некоторой окрестности N точки $(0, 0)$. За пределами этой окрестности $F(i, j)$ или в точности равно нулю, или очень близко к нему.





Ядро фильтра

- Суммирование производится по $(i, j) \in N$, и значение каждого пикселя $B(x, y)$ определяется пикселями изображения A , которые лежат в окне N , центрированном в точке (x, y) (будем обозначать это множество $N(x, y)$). Ядро фильтра, заданное на прямоугольной окрестности N , может рассматриваться как матрица m на n , где длины сторон являются нечетными числами. При задании ядра матрицей M_{kl} , ее следует центрировать:

$$F(i, j) = M_{i + \frac{m-1}{2}, j + \frac{n-1}{2}}$$





Условия на границе

Если пиксель (x, y) находится в окрестности краев изображения. В этом случае $A(x + i, y + j)$ может соответствовать пикселю A , лежащему за пределами изображения A . Данную проблему можно разрешить несколькими способами:

- Не проводить фильтрацию для таких пикселей, обрезав изображение B по краям или закрасив их, к примеру, черным цветом.
- Не включать соответствующий пиксель в суммирование, распределив его вес $F(i, j)$ равномерно среди других пикселей окрестности $N(x, y)$.
- Доопределить значения пикселей за границами изображения при помощи экстраполяции.
- Доопределить значения пикселей за границами изображения, при помощи зеркального отражения.





Сглаживающие фильтры

- Сглаживающие фильтры действуют на изображение аналогично мутному стеклу: изображение становится нерезким, размытым. Простейший *прямоугольный сглаживающий фильтр* радиуса r задается при помощи матрицы размера $(2r + 1) \times (2r + 1)$, все значения которой равны

$$\frac{1}{(2r + 1)^2}$$

- а сумма по всем элементам матрицы равна, таким образом, единице. При фильтрации с данным ядром значение пикселя заменяется на усредненное значение пикселей в квадрате со стороной $2r+1$ вокруг него.





Примеры сглаживающих фильтров

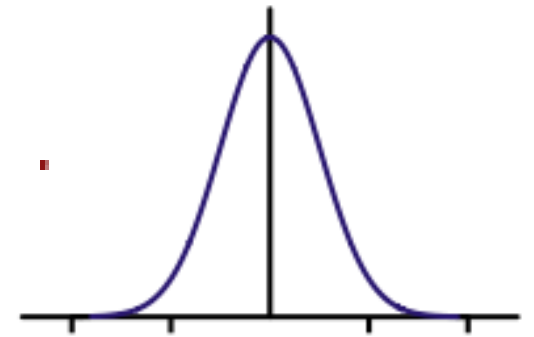




Гауссовский фильтр

- Гауссовский фильтр – это линейный фильтр имеющий следующее ядро:

$$F_{gauss}(i, j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right).$$



- Где σ^2 – дисперсия случайной величины
- Дисперсия — мера разброса данной случайной величины, то есть её отклонения от математического ожидания.
- Гауссовский фильтр имеет ненулевое ядро бесконечного размера. Однако ядро фильтра очень быстро убывает к нулю при удалении от точки $(0, 0)$, и потому на практике можно ограничиться сверткой с окном небольшого размера вокруг $(0, 0)$ (например, взяв радиус окна равным 3σ).





Контрастоповышающие фильтры

- Ядро контрастоповышающего фильтра имеет значение, большее 1, в точке (0, 0), при общей сумме всех значений, равной 1.

$$M_1^{contr} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad M_2^{contr} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

- Характерным артефактом линейной контрастоповышающей фильтрации являются заметные светлые и менее заметные темные ореолы вокруг границ.





Пример



эффект от
применения
фильтра с
ядром

M_2^{cont}





Программная реализация

```
f[1,1] := 1/9; f[1,2] := 1/9; f[1,3] := 1/9;  
f[2,1] := 1/9; f[2,2] := 1/9; f[2,3] := 1/9;  
f[3,1] := 1/9; f[3,2] := 1/9; f[3,3] := 1/9;  
with Image1.Canvas do  
for i := 1 to Image1.Picture.Width - 2 do  
  for j := 1 to Image1.Picture.Height - 2 do  
    begin  
      m[1,1]:=Pixels[i-1,j-1]; m[1,2]:=Pixels[i,j-1]; m[1,3]:=Pixels[i+1,j-1];  
      m[2,1]:=Pixels[i-1,j]; m[2,2]:=Pixels[i,j]; m[2,3]:=Pixels[i+1,j];  
      m[3,1]:=Pixels[i-1,j+1]; m[3,2]:=Pixels[i,j+1]; m[3,3]:=Pixels[i+1,j+1];  
      Image2.Canvas.Pixels[i,j]:= Mul(f,m);  
    end;
```





Программная реализация

```
function Mul(var f: masR; var m: masI): TColor;  
var i,j,k: Integer;  s: Real;  
begin  
    for i:=1 to 3 do  
        for j:=1 to 3 do  
            m[i,j] := GetValue(m[i,j]);  
S := 0;  
    for i:=1 to 3 do  
        for j:=1 to 3 do  
            S := S + f[i,j]*m[i,j];  
k := round(S); Result := RGB(k,k,k);  
end;
```





Разностные фильтры

- В отличие от сглаживающих и контрастоповышающих фильтров, не меняющих среднюю интенсивность изображения (сумма элементов ядра равна единице), в результате применения разностных фильтров получается, как правило, изображение со средним значением пикселя близким к нулю. Вертикальным перепадам (границам) исходного изображения соответствуют пиксели с большими по модулю значениями на результирующем изображении. Поэтому разностные фильтры называют также **фильтрами, находящими границы**
- Фильтры Прюита (Prewitt) и Собеля (Sobel):

$$M_1^{prewitt} = \frac{1}{3} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} . \quad M_1^{sobel} = \frac{1}{4} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} .$$



Пример



- серый цвет соответствует значению 0
- **Недостатки:** неопределенность в выборе величины порога. Для разных частей изображения приемлемый результат обычно получается при существенно разных пороговых значениях. Кроме того, разностные фильтры очень чувствительны к шумам изображения.





Нахождение границ по Собелю

- Пусть A исходное изображение, а G_x и G_y - два изображения, где каждая точка содержит приближенные производные по x и по y .

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A \quad \text{and} \quad G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

где $*$ обозначает двумерную операцию свертки (операцию линейной фильтрации, рассмотренную ранее).

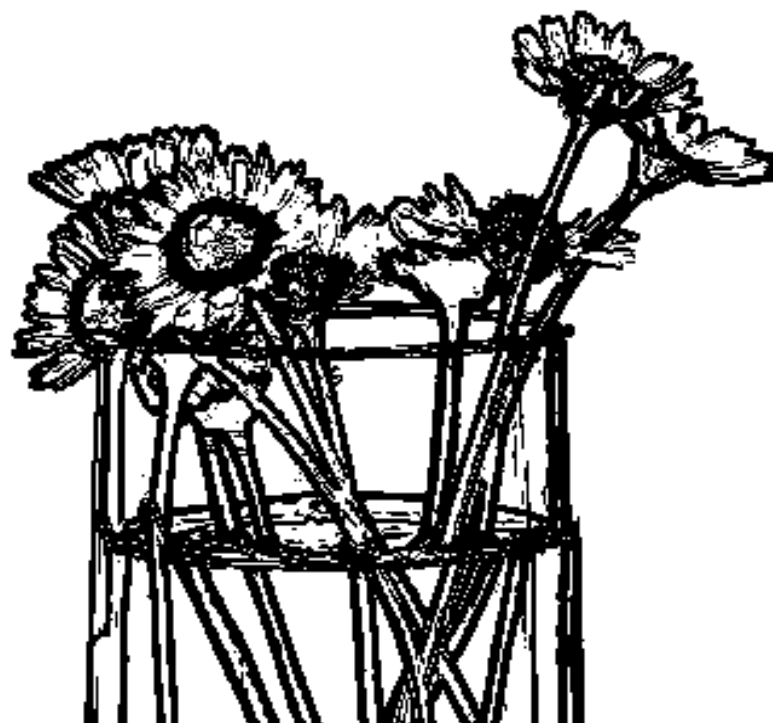
В каждой точке изображения приближенное значение величины градиента можно вычислить, используя полученные значения G_x и G_y

$$G = \sqrt{G_x^2 + G_y^2}$$





Использование фильтра Собеля





Нелинейные фильтры

- Одним из примеров нелинейного фильтра является пороговая фильтрация. Результатом пороговой фильтрации служит бинарное изображение, определяемое следующим образом:

$$B(x, y) = \begin{cases} 1, & \text{если } A(x, y) > \gamma \\ 0, & \text{иначе} \end{cases} .$$

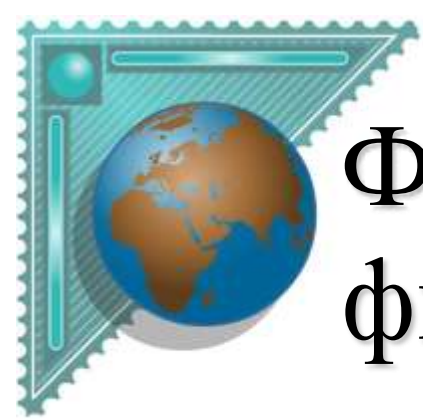




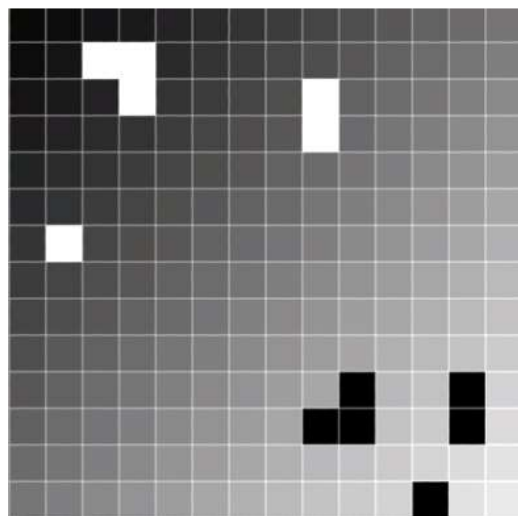
Медианный фильтр

- Как и в линейных фильтрах, по пикселям передвигается окно, которое охватывает пиксели, участвующие в формировании итоговой интенсивности.
- Значения внутри этого окна воспринимаются как одномерный массив, который сортируется в порядке возрастания.
- Значение, находящееся в середине отсортированного массива, поступает на выход фильтра.

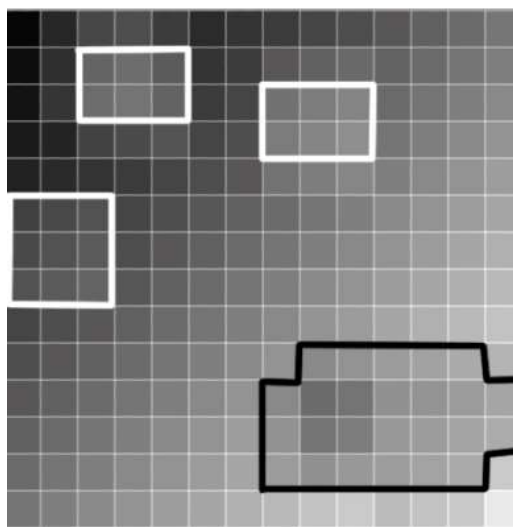




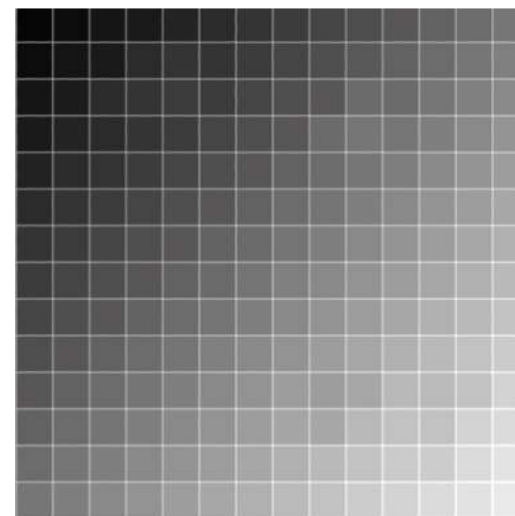
Фильтры размытия и медианный фильтр



а)



б)



в)

