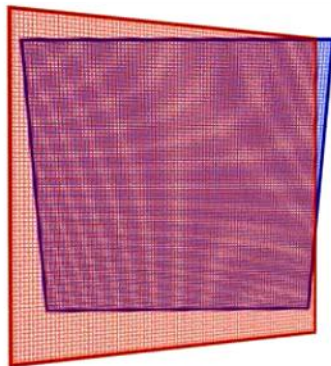
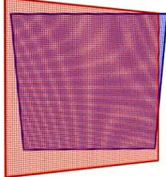


## ТЕМА №21

# Щриховане и шлифоване



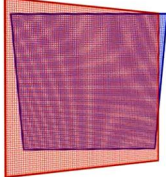


# Съдържание

## Тема 21: Щриховане и шлифоване

- Щриховане (*dithering*)
- Шлифоване (*antialiasing*)

**Щриховане**



# Етимология

## Щрих

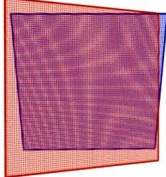
- От немски *strich* – линия, черта

## Най-често

- Къса и тънка чертичка

## Още по-най-често

- В съседство с други подобни линии



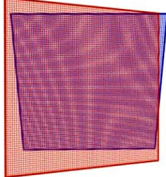
# Но на английски

## Официалният термин

- От английски *dithering* – *треперя, колеба*

## На български

- Често се превежда като „разпространяване на грешката“, но това всъщност е само един от класовете алгоритми за щриховане



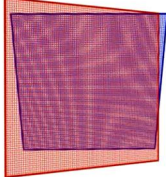
# Цел на щриховането

## Основна цел

- Представяне на различен интензитет на изображение чрез щрихи
- Щрихите са с фиксиран цвят и интензитет

## Цветът се постига чрез

- Промяна на дебелината на щрихите
- Промяна на гъстотата на щрихите

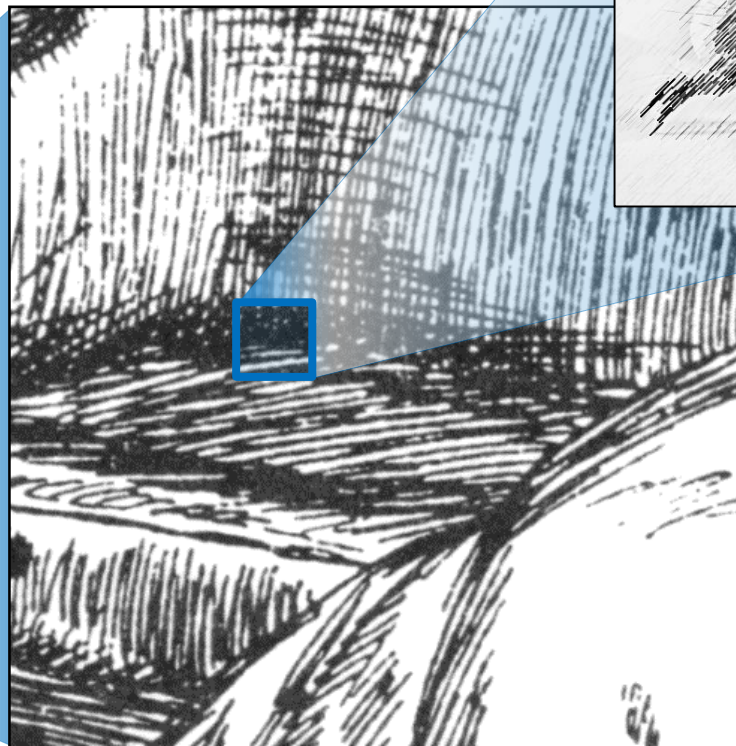


# Използване

## В изкуството

- Полутонове с едно и също мастило
- Относително рядко в картини
- Много по-често при илюстрации в книги поради техническо недостъпен полутон и цветен печат
- Днес – за имитация на старинен стил

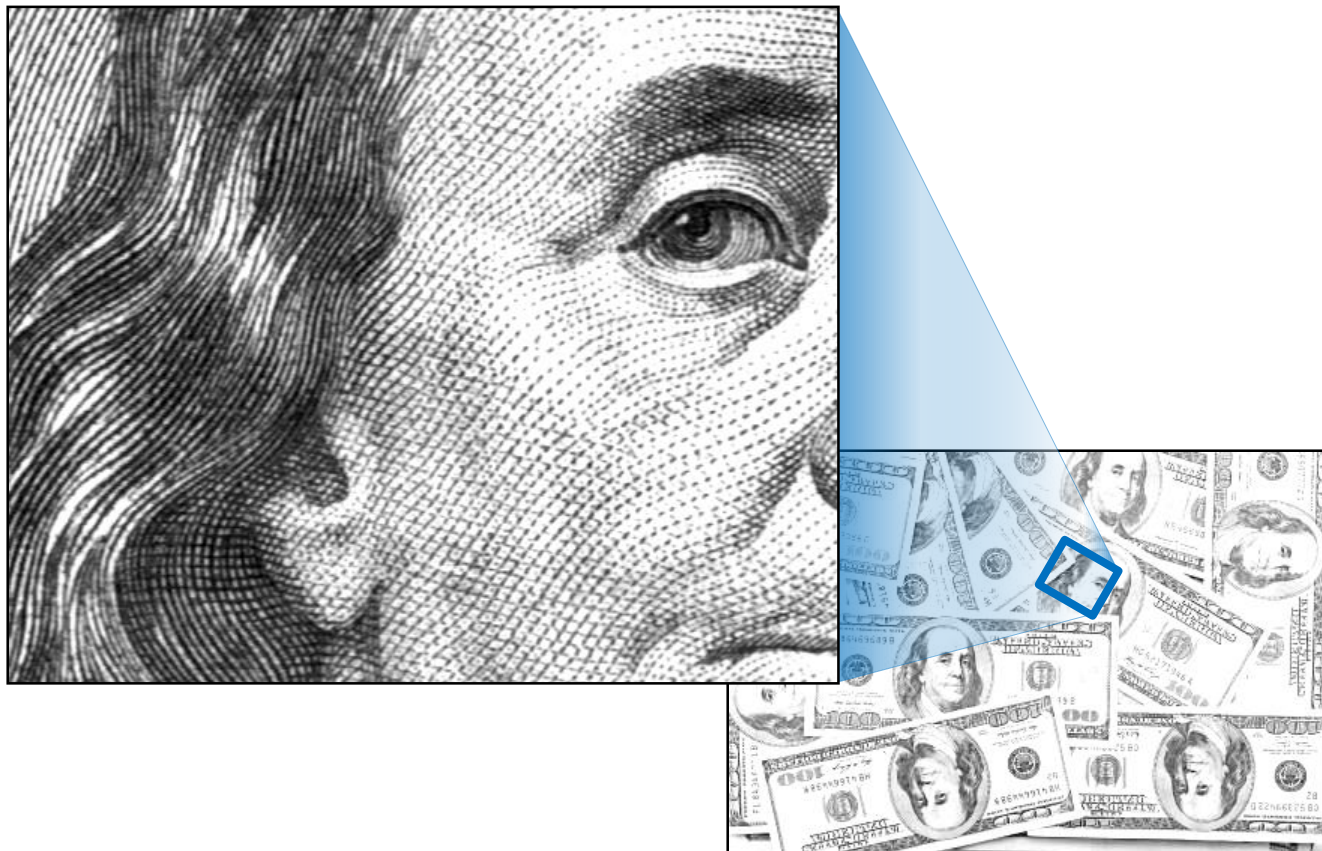
# Пример с илюстрация



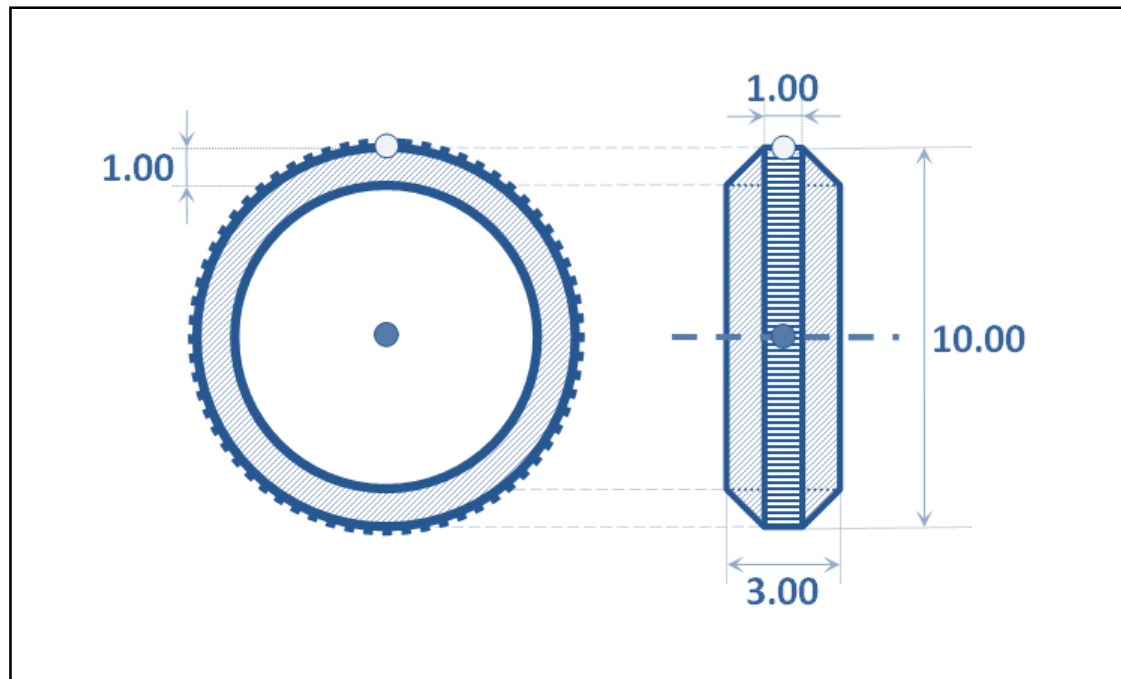
Илюстрация на Хю Томсън „Той изрязва дълга къдрица от косата ѝ“  
от книгата на Джейн Остен „Чувство и чувствителност“

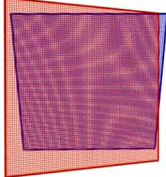


# Пример с банкнота US\$100



# Пример с чертеж

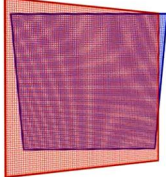




# В компютърната графика

## Щриховането в КГ

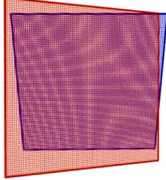
- Прилага се при конвертиране на полутонови изображения в черно-бели
- Различната степен на интензитет от черно до бяло се представя чрез комбинации само от черно и бяло
- Щриховането е чрез пиксели, а не чрез щрихи (изключение са някои филтри)



# Обобщение на щриха

## Щриховане се прилага

- При представяне на пълноцветни изображения чрез краен, често малък брой цветове
- Подходящо при конвертиране до GIF, PNG8 или до изображения с фиксирани палитри



# Брой цветове

## Най-масовите изображения

- Имат 24-битови цветове: по 8 бита за червена, зелена и синя компоненти на цвета на всеки пиксел
- Това е почти 17 милиона цвята

## При намаляване на броя на цветовете

- Трябва да жертваме някои
- Кой и как – това се решава с dithering

# Пример

17 милиона цвята



64 цвята

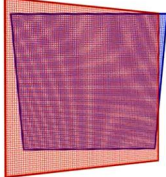


8 цвята



3 цвята





# Щриховане с два цвята

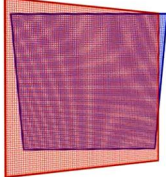
## Щриховане с черно и бяло

- Кой точно пиксели да са бели и кои не
- Максимална привлекателност

## Решения

- С граница на интензитета
- С готови шаблони
- С интензитетен шум
- С разпространение на грешката





# Граница на интензитета

## Алгоритъм

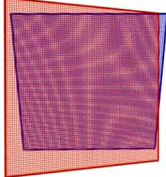
- Определя се степен на интензитета
- Всички по-тъмни пиксели стават черни
- Всички по-светли стават бели

## Особености

- На английски се казва *thresholding*
- Много бърз алгоритъм, но с загуба на детайли

# Пример





# Намиране на границата

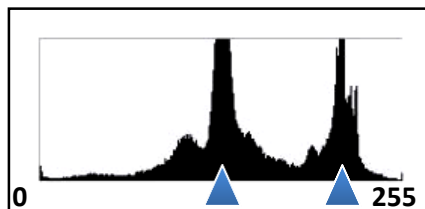
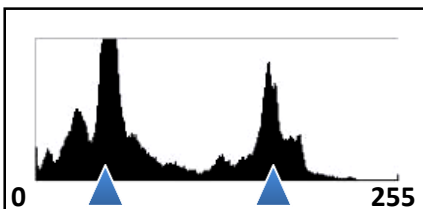
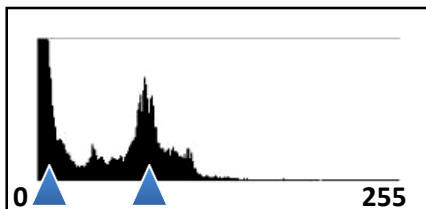
## Наивен метод

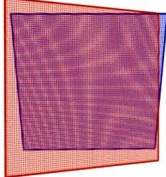
- Абсолютната среда (интензитет 127.5)

## Чрез хистограма

- Разглежда се броят пиксели от всеки интензитет:  
 $n_0, n_1, \dots, n_{255}$
- Границата е „център на масата“
- Двата цвята на щриха може да не са бял и черен

# Хистограма





# Щриховане с шаблони

## Набор от шаблони $2 \times 2$ , $3 \times 3$ , ...

- Фиксира се разпределение на бели и черни пиксели
- На всеки шаблон отговаря диапазон от интензитета

## Алгоритъм

- Изображението се разбива на части
- Всяка част се подменя с шаблон

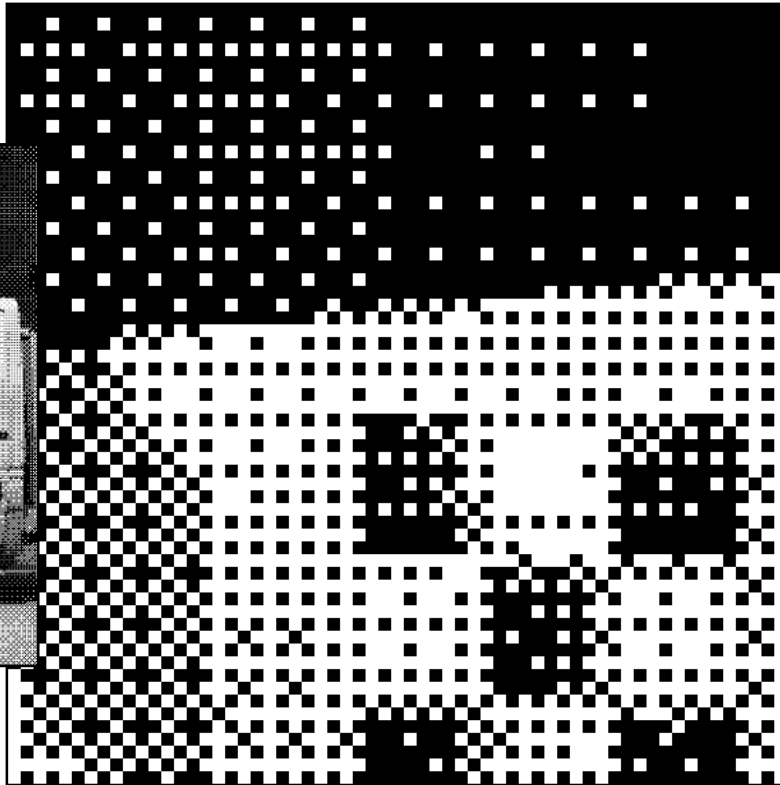
## **Предимства**

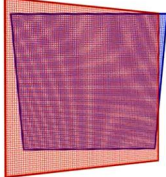
- Придава по-точно интензитета
- Показва по-добре детайлите
- Относително бърз алгоритъм

## **Недостатъци**

- Шаблоните придават неестественост, породена от специфичната им подредба

# Пример

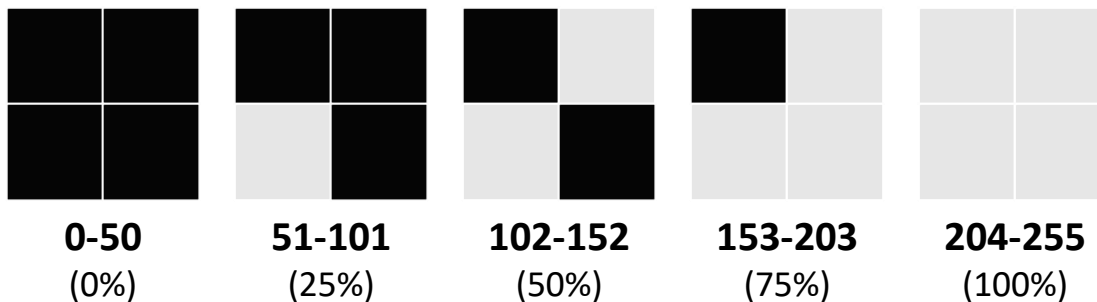




# Шаблони

## Примерен комплект шаблони 2×2

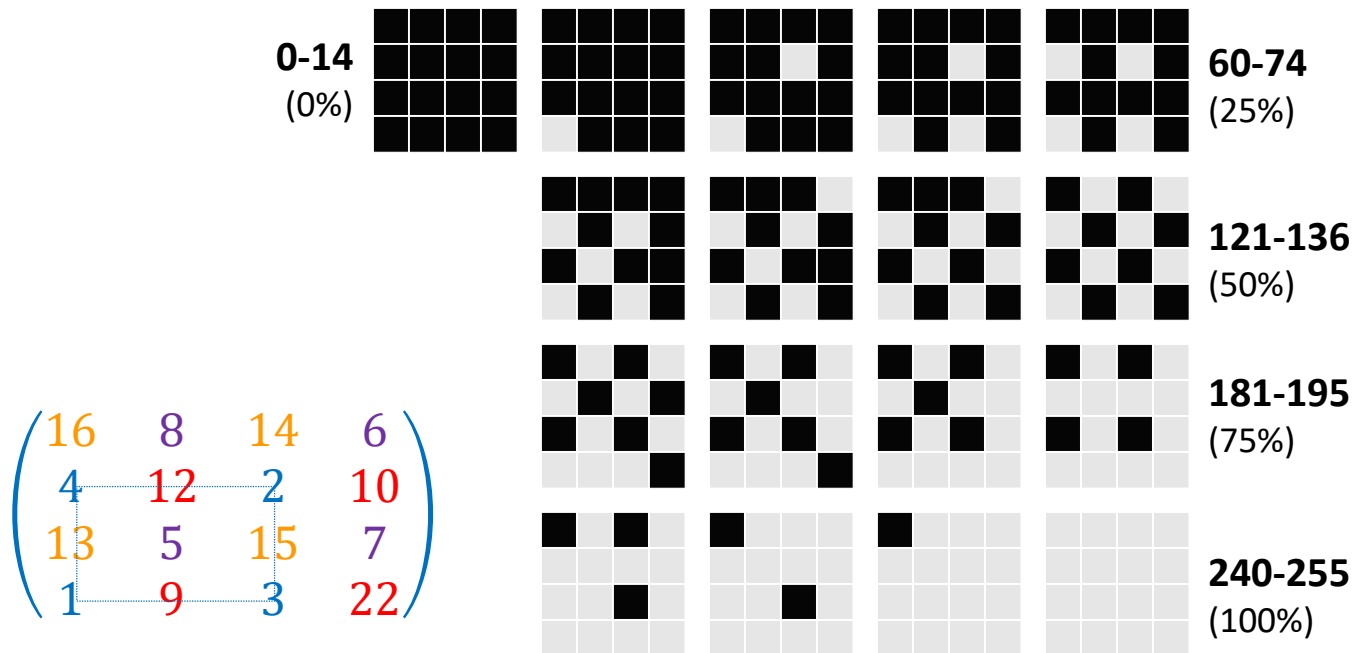
– Пет шаблона

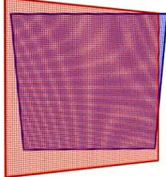


– Матрична дефиниция  $\begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}$   
(т.е. ред на запълване)



# Примерен комплект шаблони 4x4





# Интензитетен шум

## Шум (случаен цвят на пиксел)

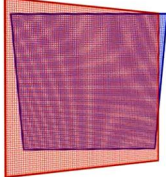
- Интензитетът на пиксел определя вероятността да стане бял или черен

## Особености

- Положителна: елиминира шаблонните артефакти
- Отрицателна: получава се прекалено хаотична за човек щриховка

# Пример



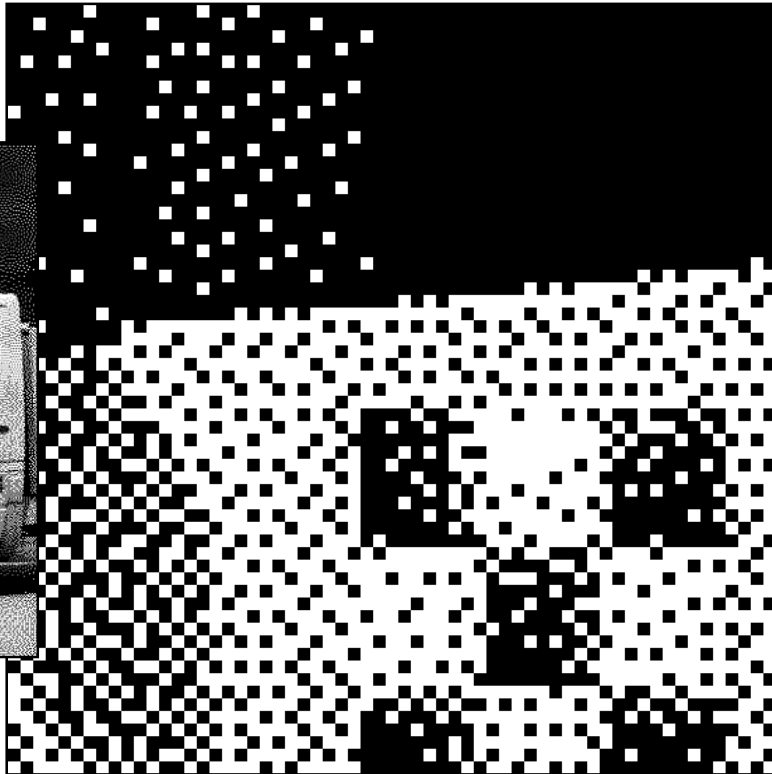


# Разпространение на грешката

## Основна идея

- След замяната на цвета на пиксел, се намира разликата до оригиналния цвят на пиксела
- Тази разлика се нарича „грешка“
- Грешката се разпределя из пикселите под и вдясно от този пиксел
- Най-известен алгоритъм е този на Флойд-Щайнберг

# Пример

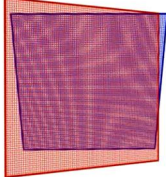


## **Предимства**

- Пикселите са без шаблон и почти няма артефакти
- Детайлите са видими

## **Недостатъци**

- Най-бавен метод за щриховане  
(спрямо предишните три)



# Алгоритъм

## Алгоритъм на Флойд-Щайнберг

- Пикселите се обхождат надясно и надолу
- Интензитетът на пиксел се подменя с най-близкия допустим интензитет
- Разликата между двата интензитета („грешката“) се разпределя в пикселите под и вдясно от текущия пиксел

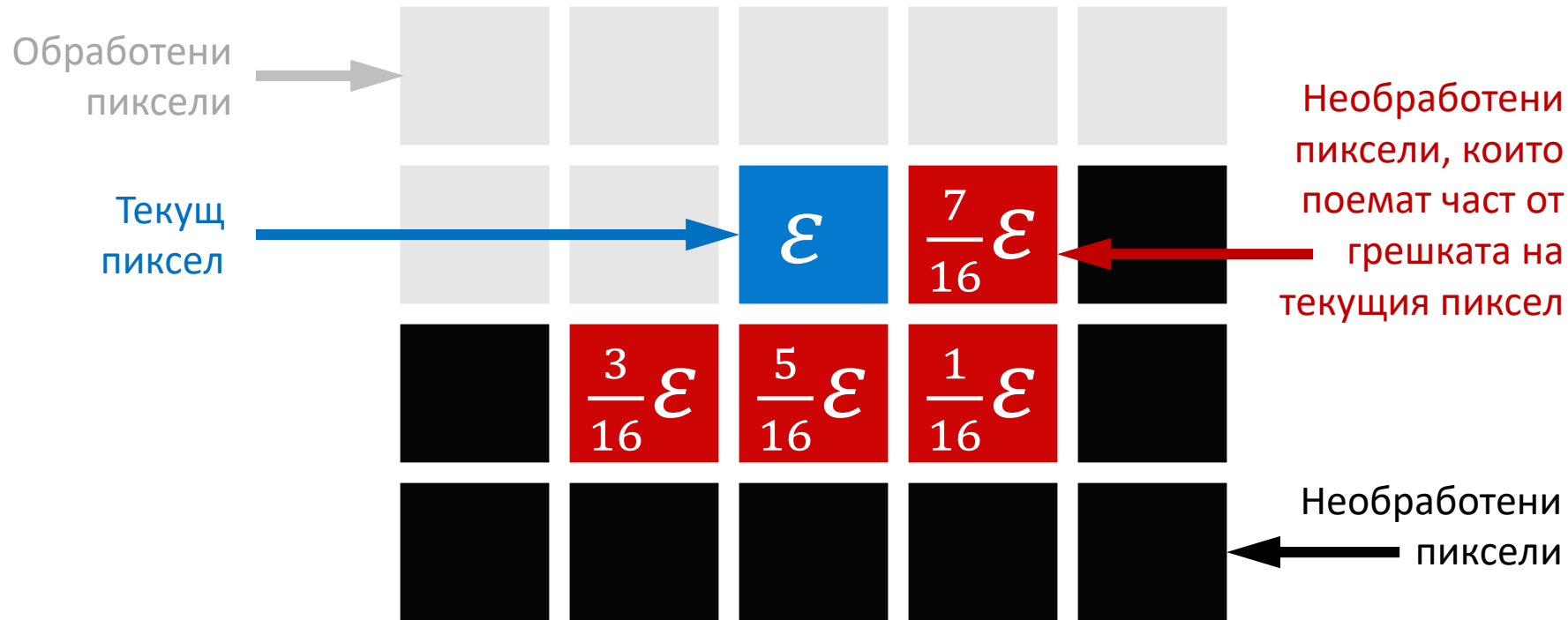
## Ефект от разпределението

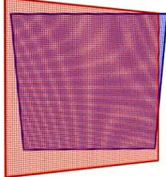
- Ако пиксел е станал по-светъл, отколкото е бил оригинално, за околните пиксели е по-вероятно да станат по-тъмни
- Разпределението на грешката е такова, че сумарната грешка от всички пиксели е почти нула

## Разпределение на грешката

- Извършва се от проста схема
- Ако грешката е  $\varepsilon$ , то схемата е:







# Псевдокод

За всяко  $Y$  отгоре-надолу

За всяко  $X$  отляво-надясно

стар  $\leftarrow$  пиксел( $X, Y$ )

нов  $\leftarrow$  нов-интензитет(стар)

$\varepsilon \leftarrow$  стар-нов

пиксел( $X, Y$ )  $\leftarrow$  нов

пиксел( $X+1, Y$ )  $+= 7/16 * \varepsilon$

пиксел( $X-1, Y+1$ )  $+= 3/16 * \varepsilon$

пиксел( $X, Y+1$ )  $+= 5/16 * \varepsilon$

пиксел( $X+1, Y+1$ )  $+= 1/16 * \varepsilon$

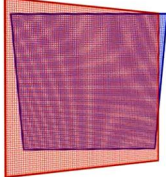
Това е тире

А това - минус

Това също  
е тире

На кого  
2 точки?

**Шлифоване**



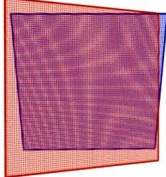
# Етимология

## Шлифовам, шлайфам

- От немски *schliff, schleif* – изрязвам, шлайфам

## Обаче

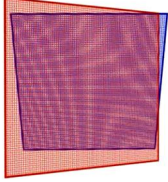
- На английски: *antialiasing*
- В обработката на сигнали *aliasing* е дефект от дигитализацията или понижаването на честотата на сигнала



# В компютърната графика

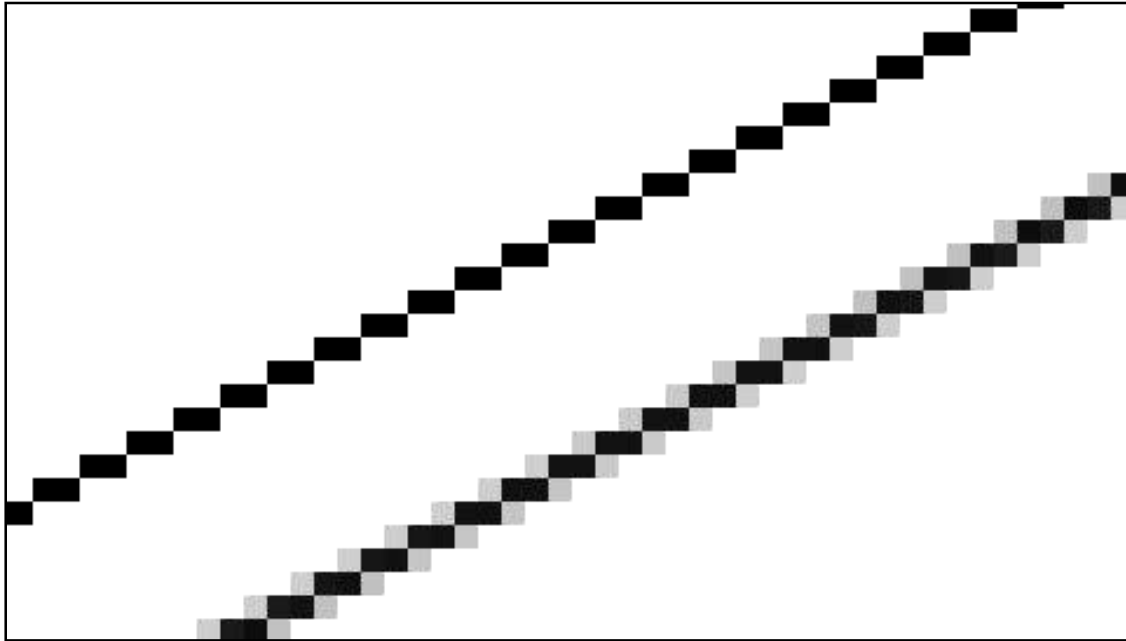
## Проблеми в КГ

- При растеризация на почти всички примитиви (дори линиите и точките)
- Наклонените линии изглеждат „на стълбички“
- Извивките на буквите изглеждат „ръбести“

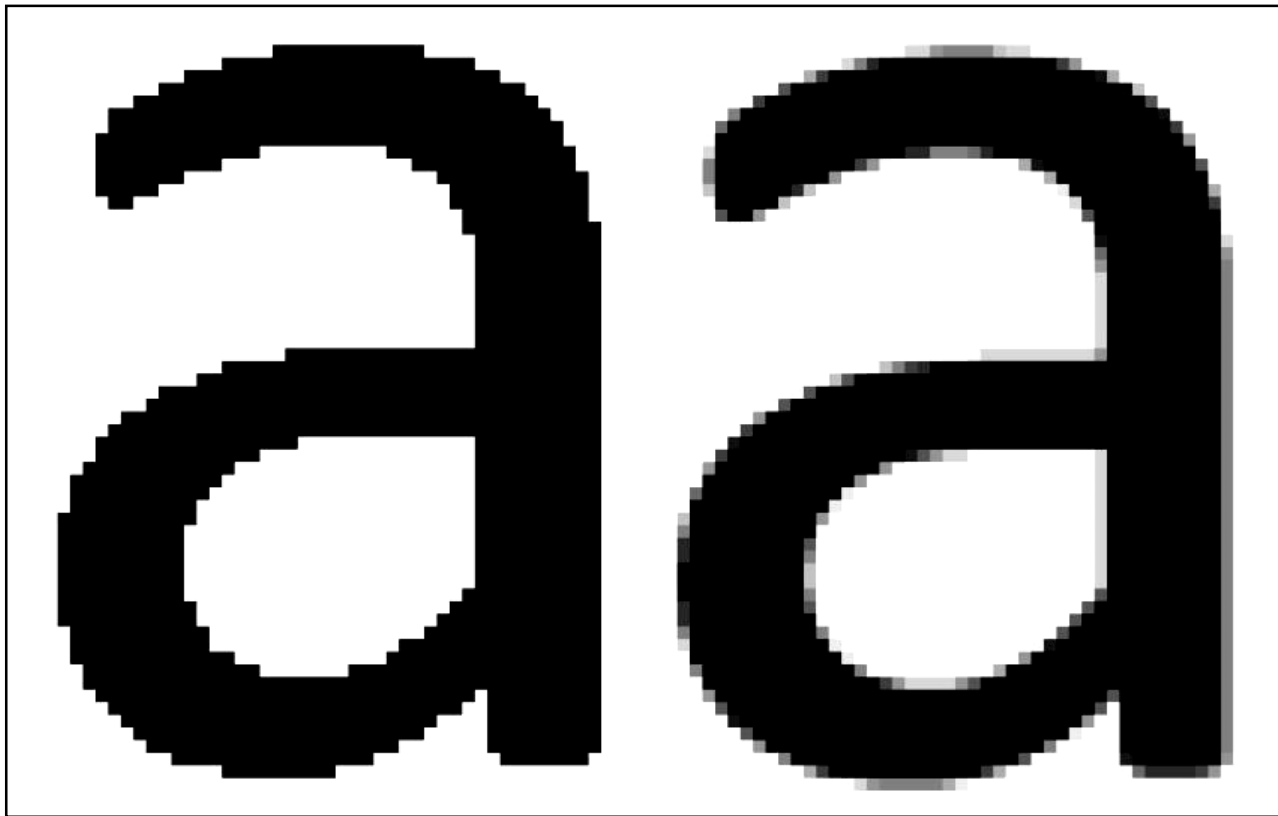


# Примери

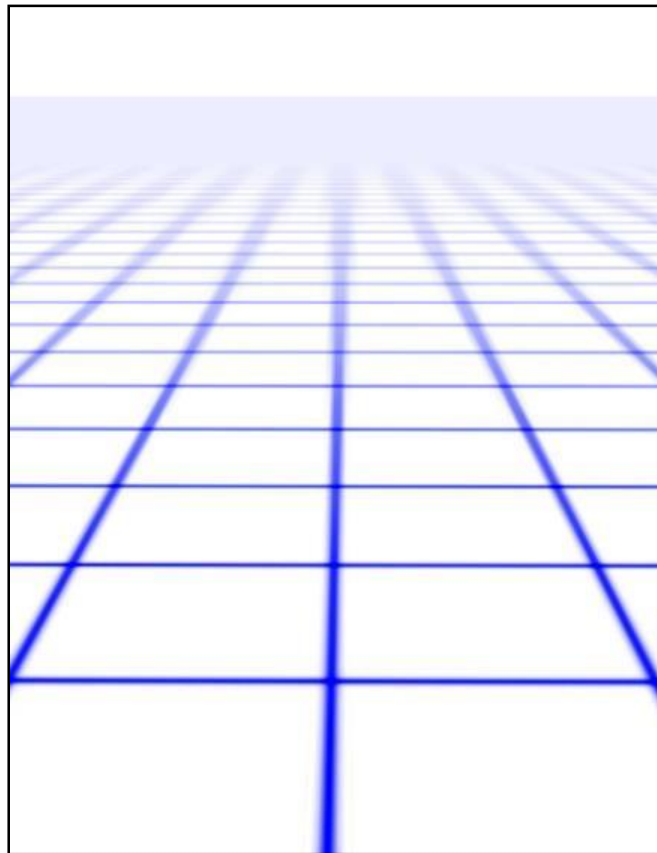
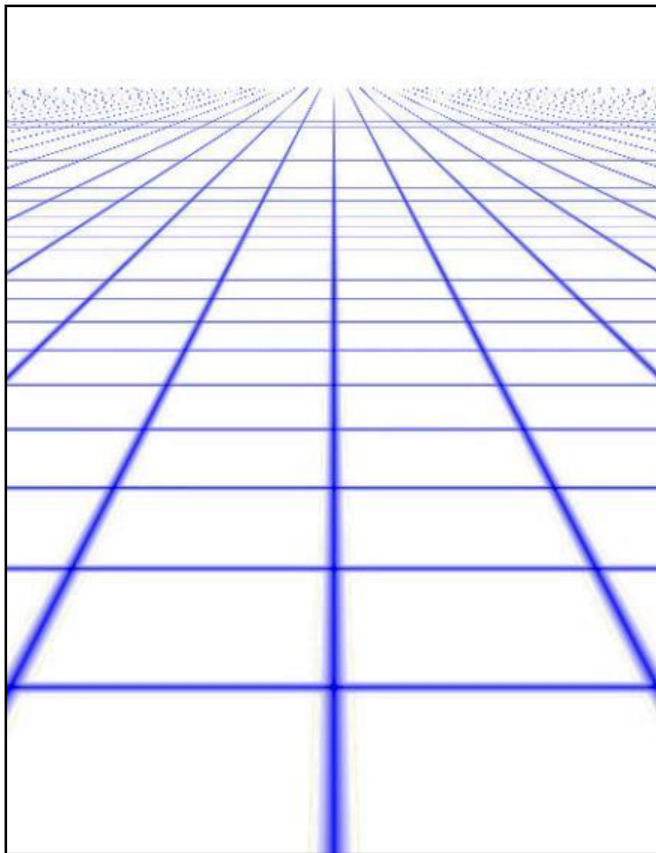
Изглаждане при линии



Изглаждане при символи

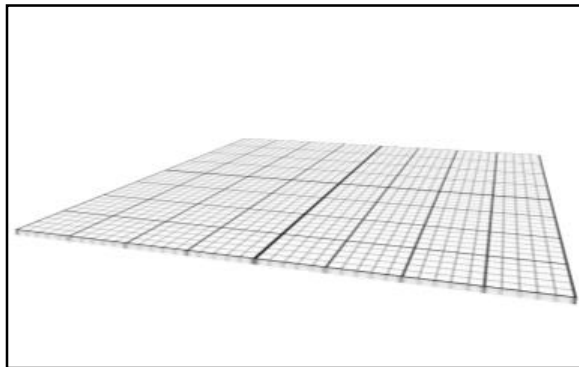


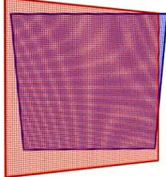
# Заглаждане при текстури





**Да го видим**





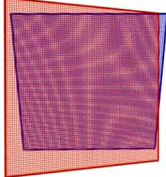
# Как се изглажда

## Алгоритми за изглаждане

- Чрез работа на подпикселно ниво
- Чрез оценка на припокрита площ
- Чрез разстояние до примитива

## В тази лекция

- Ще разгледаме първия алгоритъм
- Останалите вижте в допълнителната литература (най-вече в [LUKI])



# Подпикселно ниво

## Основна идея

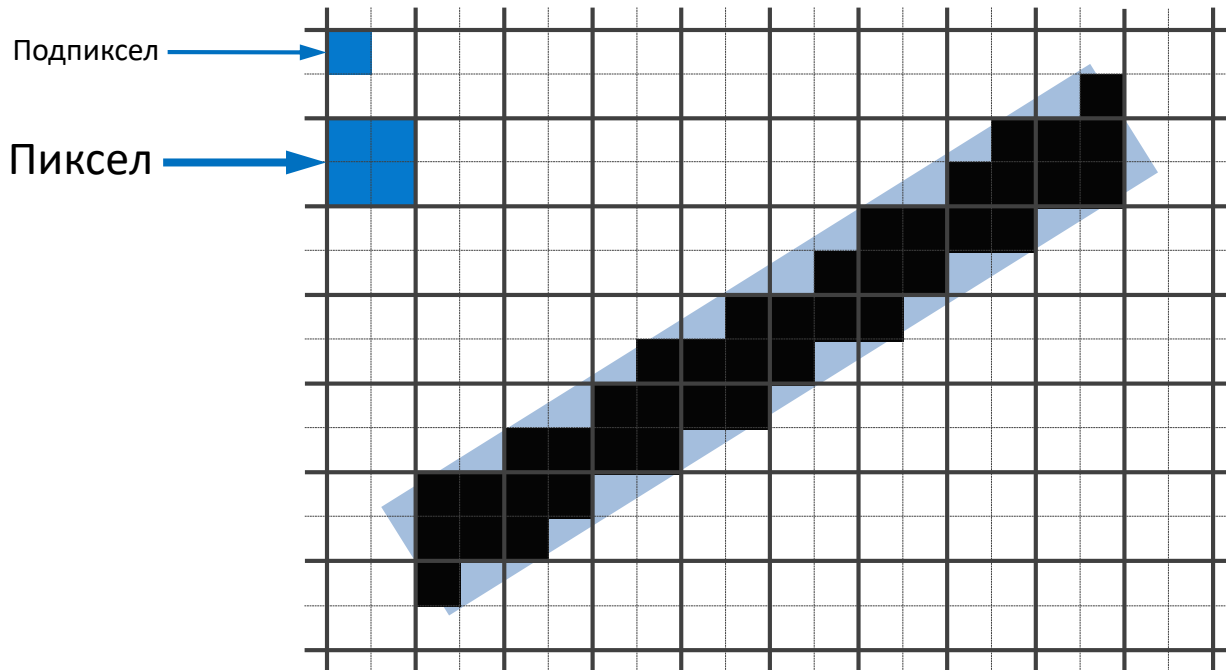
- Всеки пиксел се разбива на мрежа от по-малки *подпиксели (subpixels)*
- Растеризирането се извършва с подпикселите
- Интензитетът на оригиналните пиксели се определя от броя бели и черни подпиксели



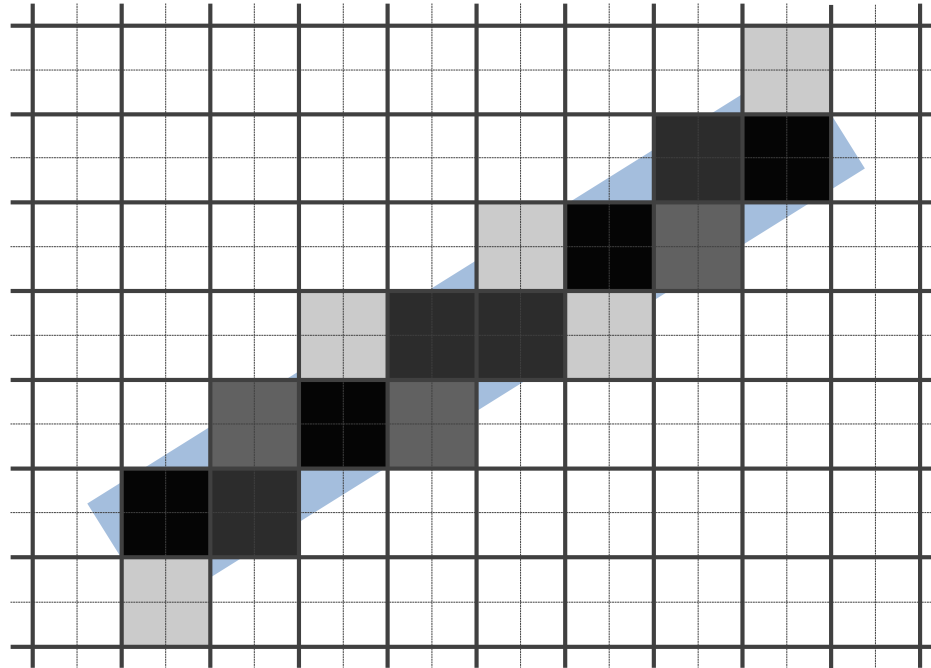
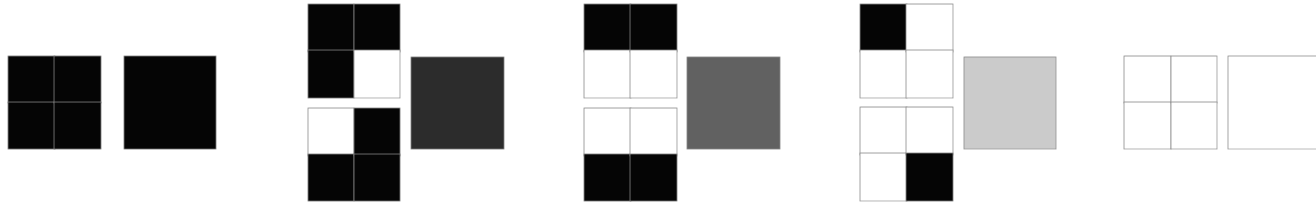
The diagram shows a sequence of black squares on a grid, connected by a light blue shaded area. The squares are arranged in a diagonal pattern from the bottom-left to the top-right. The shaded area follows the path of the squares, indicating a continuous sequence or path.

# Растеризация на ниво подпиксели

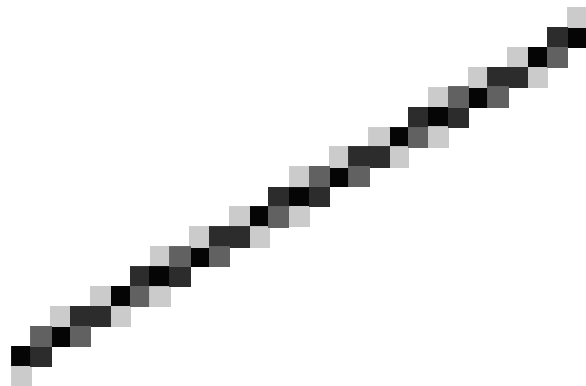
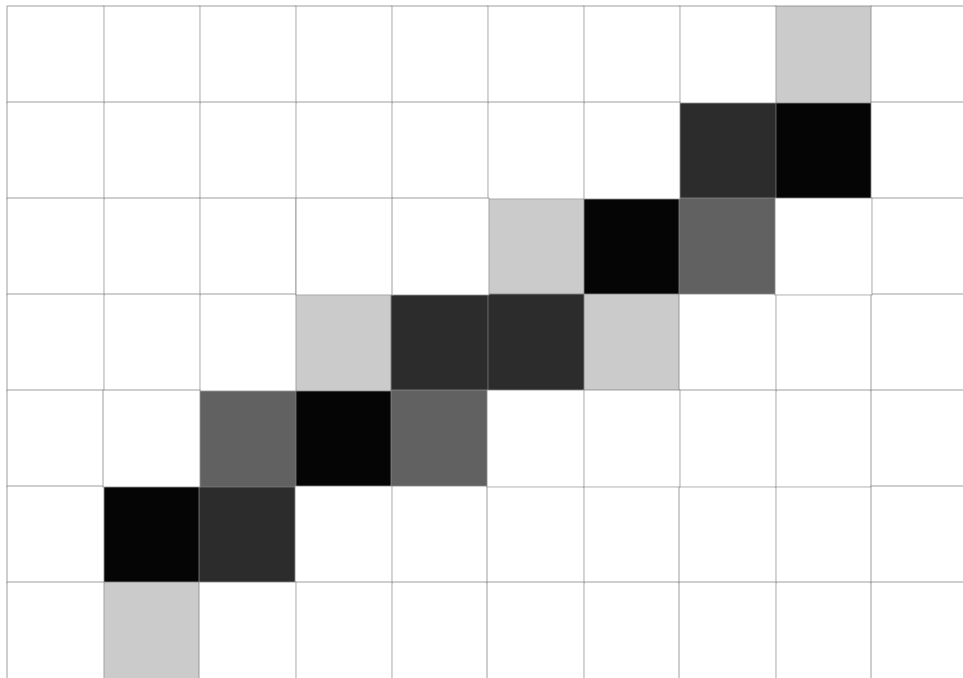
- Растеризира се линия с дебелина
- Един пиксел =  $2 \times 2$  подпиксела



# На ниво пиксели - 5 степени на сивост

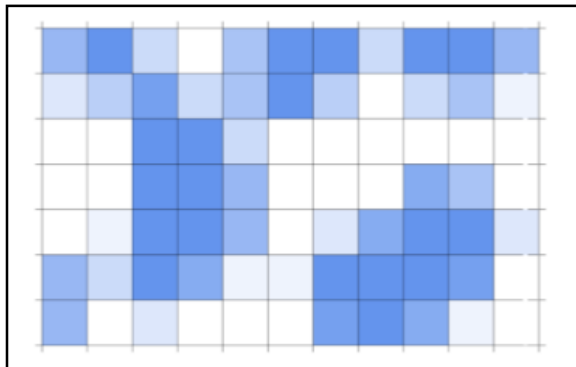


# Краен резултат



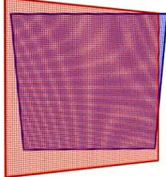
# Демонстрация

- Изглаждане с подпиксели  $3 \times 3$





**Въпроси?**



# Повече информация

- [[LUKI](#)]      стр. 69-75
- [[AGO2](#)]      стр. 48
- [[KLAW](#)]      стр. 70-74, 87-89
- [[SEAK](#)]      стр. 60-61, 165

## А също и:

- [Floyd-Steinberg Dithering](http://research.cs.wisc.edu/graphics/Courses/559-s2004/docs/floyd-steinberg.pdf)  
<http://research.cs.wisc.edu/graphics/Courses/559-s2004/docs/floyd-steinberg.pdf>
- [Anti-Aliased Line Drawing](http://courses.engr.illinois.edu/ece390/archive/archive-f2000/mp/mp4/anti.html)  
<http://courses.engr.illinois.edu/ece390/archive/archive-f2000/mp/mp4/anti.html>

**Край**