

СОФИЙСКИ УНИВЕРСИТЕТ  
„СВ. КЛИМЕНТ ОХРИДСКИ“



# Щриховане и шлифоване

ТЕМА №21

# 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

# **На английски е официалният термин**

- На английски *dithering* – треперя, колебая

# **На български**

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

## **Основна цел**

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

## **Цветът се постига чрез**

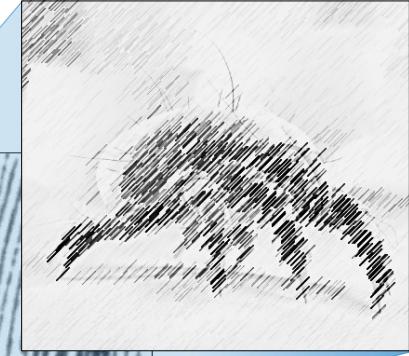
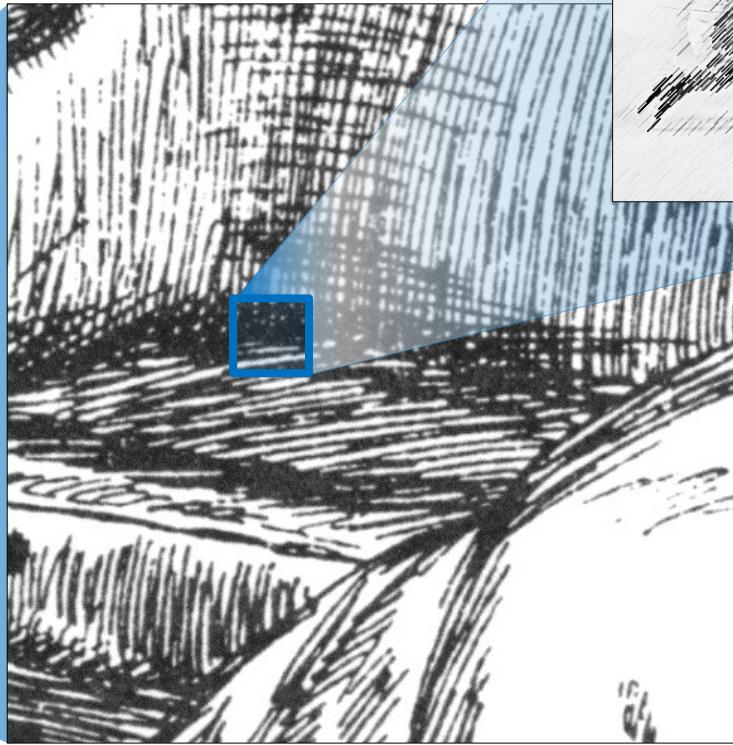
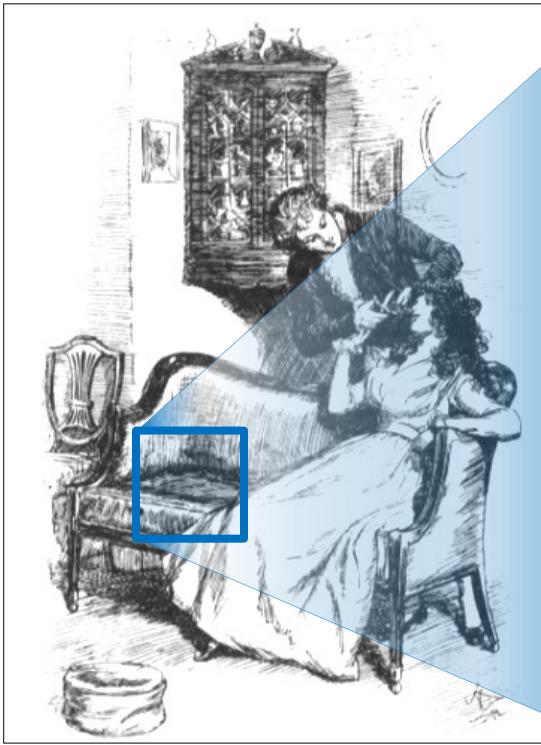
- Промяна на дебелината на щрихите
- Промяна на гъстотата на щрихите

# Използване

## В изкуството

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

# Пример с илюстрация

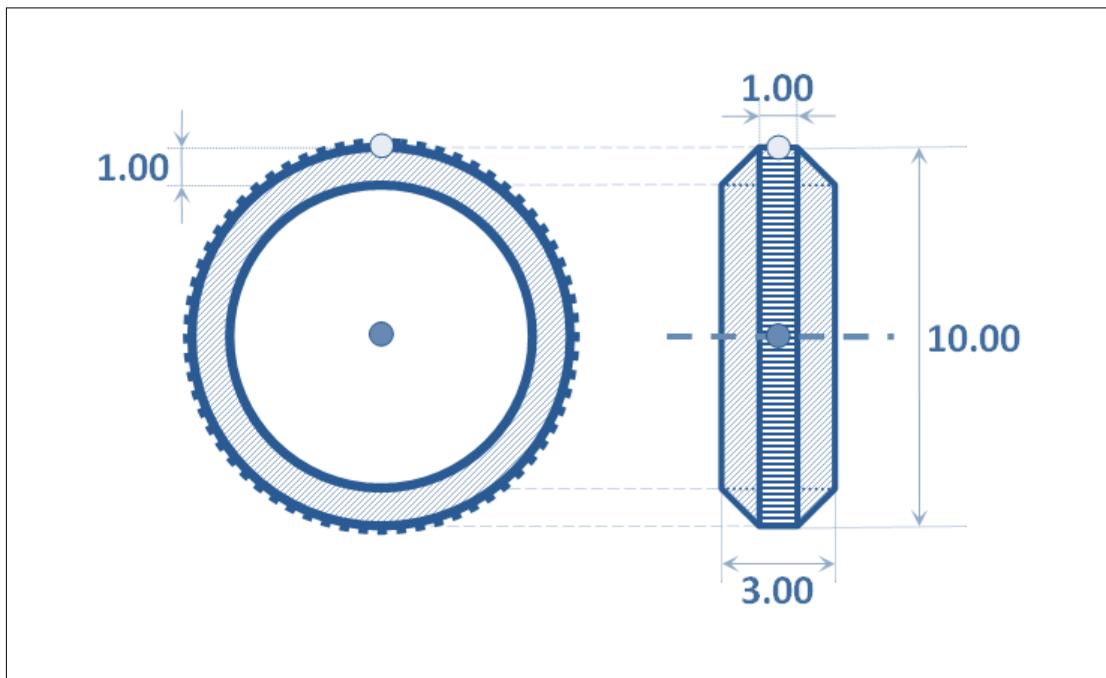


Илюстрация на Хю Томсън „Той изряза дълга къдрица от косата ѝ“  
от книгата на Джейн Остен „Чувство и чувствителност“

# Пример с банкнота US\$100



# Пример с чертеж



# **Щриховането в КГ**

- Конвертиране на полуточкови 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)

# **Щриховане се прилага**

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

## При намаляване на броя на цветовете

- Трябва да се жертват някои цветове
- Кои и как – това се решава с dithering

# Пример

17 миллиона цвята



64 цвята



8 цвята



3 цвята



# Щриховане с два цвята

## Щриховане с черно и бяло

- Кои 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



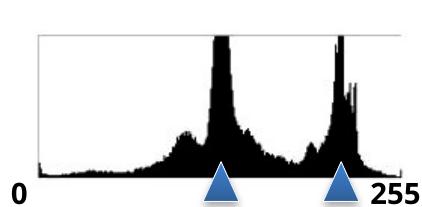
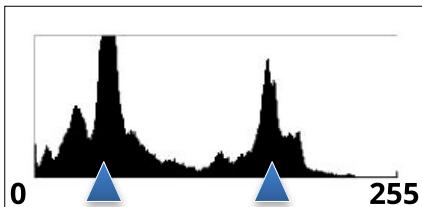
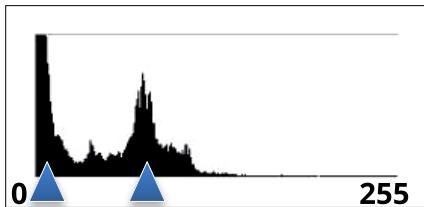
## **Намиране на границата с наивен метод**

- 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“
- Двета цвята на щриха може да не са бял и черен

# Хистограмма



# 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

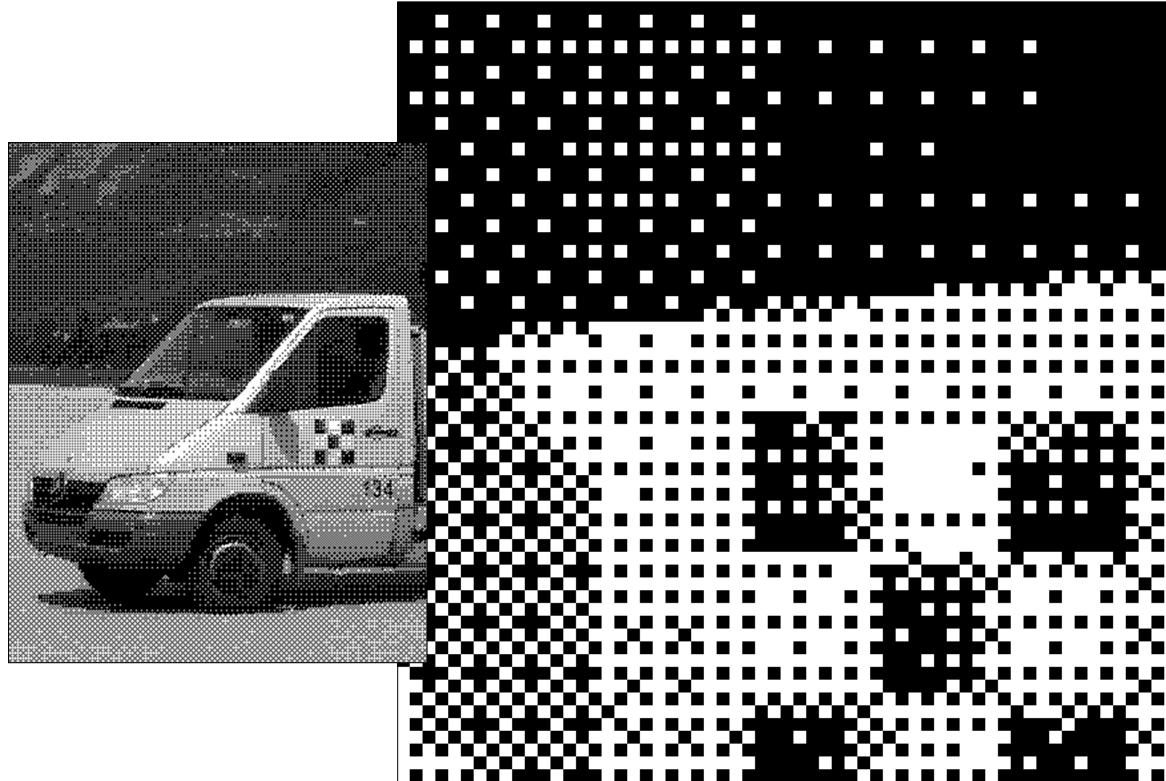
# **Предимства**

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

# **Недостатъци**

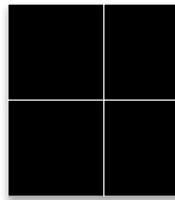
- Шаблоните придават неестественост, породена от специфичната им подредба

# Пример

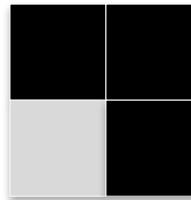


# Примерен комплект шаблони 2×2

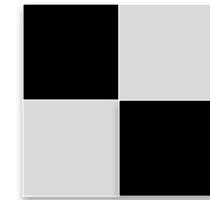
- Пет шаблона



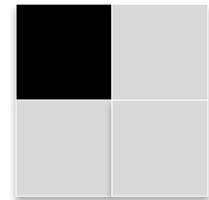
**0-50**  
(0%)



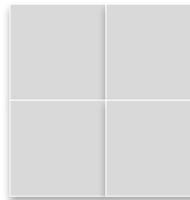
**51-101**  
(25%)



**102-152**  
(50%)



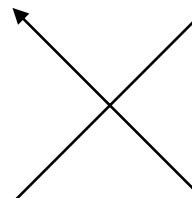
**153-203**  
(75%)



**204-255**  
(100%)

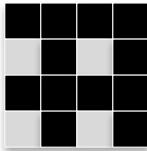
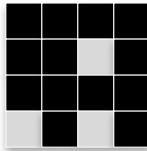
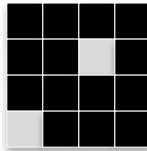
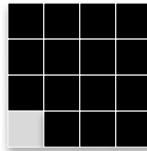
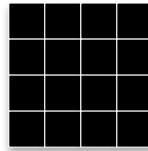
- Матрична дефиниция  
(т.е. ред на запълване)

$$\begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}$$



# Примерен комплект шаблони 4x4

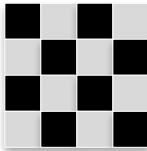
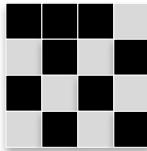
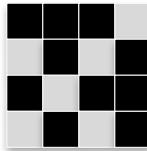
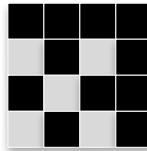
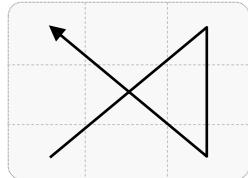
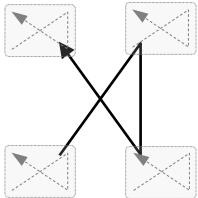
**0-14**  
(0%)



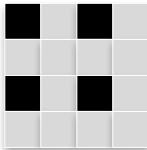
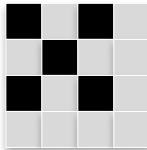
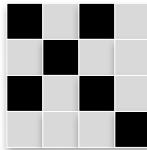
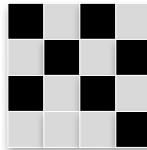
**60-74**  
(25%)

На кого 2 точки?

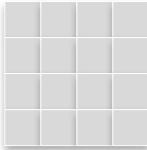
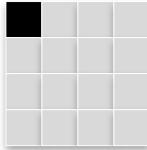
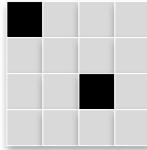
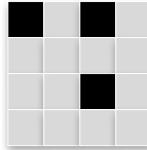
16	8	14	6
4	12	2	10
13	5	15	7
1	9	3	22



**121-136**  
(50%)



**181-195**  
(75%)



**240-255**  
(100%)

# Интензитетен шум

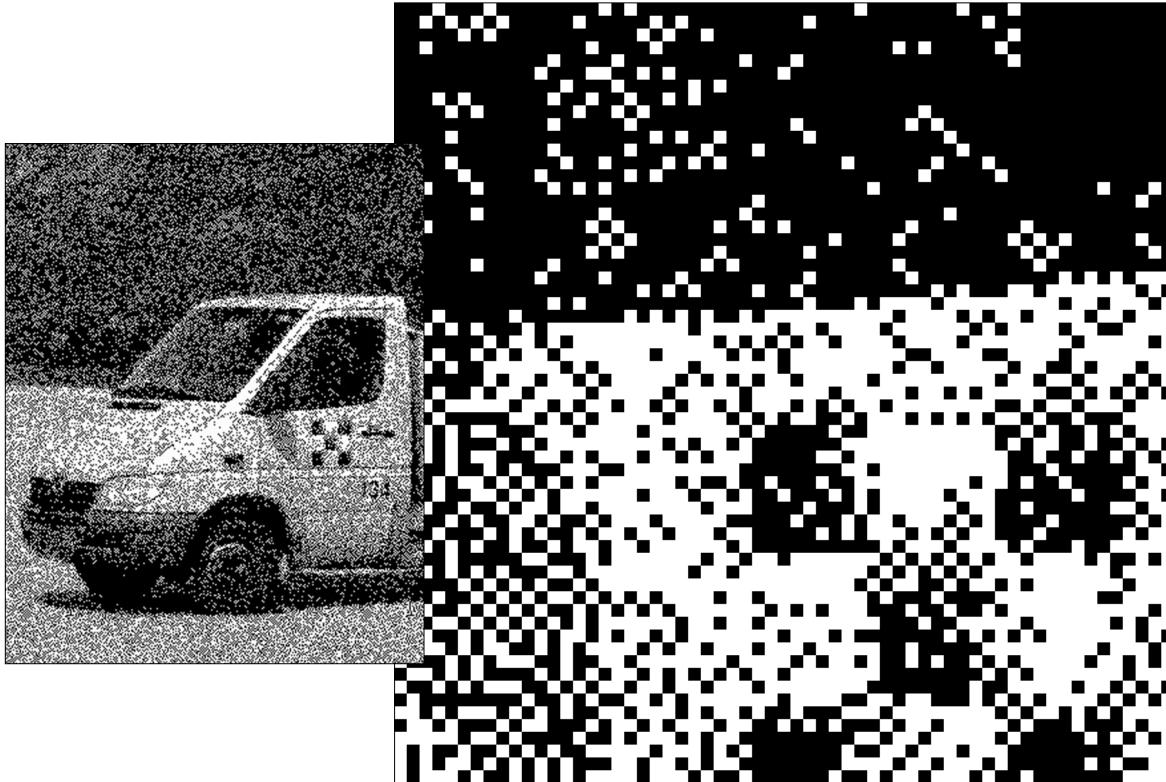
## Шум (случаен цвят на пиксел)

- Интензитетът на пиксел определя вероятността да стане бял или черен

## Особености

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

# Пример

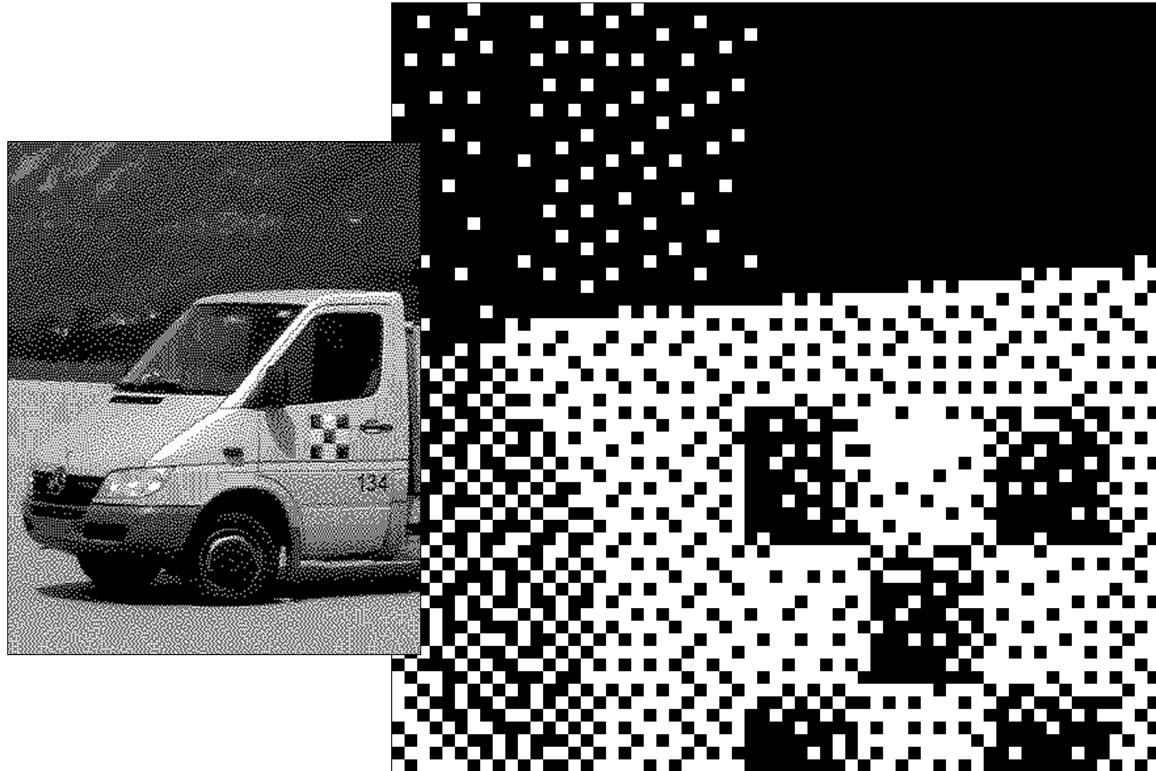


# Разпространение на грешката

## 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 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

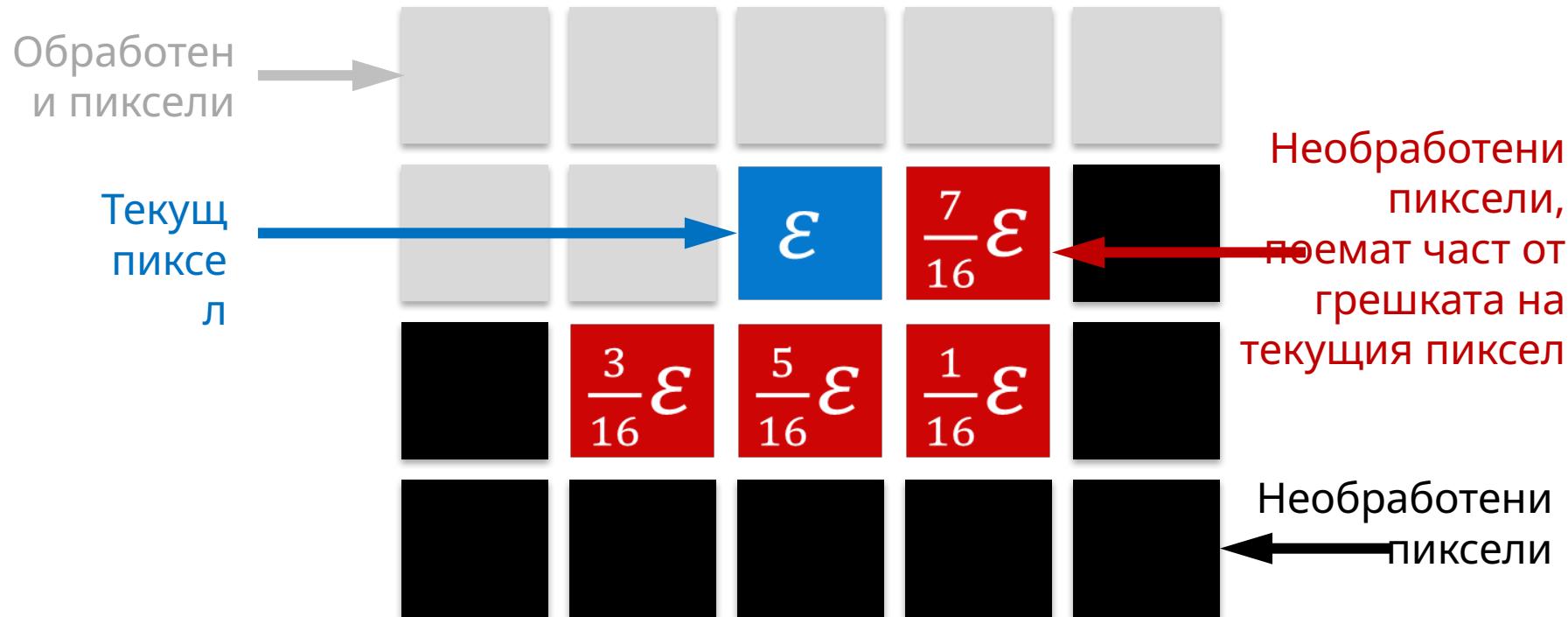
## **Разпределение на грешката**

- Извършва се от проста схема

# 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

# Ако грешката е $\varepsilon$ , то схемата е:



# Псевдокод

**За всяко Y отгоре-надолу**

**За всяко Y отляво-надясно**

- стар  $\leftarrow$  пиксел(X,Y)
- нов  $\leftarrow$  нов-интензитет(стар)
- $\epsilon$   $\leftarrow$  стар-нов
- пиксел(X,Y)  $\leftarrow$  нов
- пиксел(X+1,Y)  $+ =$   $7/16 * \epsilon$
- пиксел(X-1,Y+1)  $+ =$   $3/16 * \epsilon$
- пиксел(X, Y+1)  $+ =$   $5/16 * \epsilon$
- пиксел(X+1,Y+1)  $+ =$   $1/16 * \epsilon$

Това е тире

А това – минус

Това също е тире

А на кого 2 точки, за  
коментара, че двойният цикъл  
е все по Y?

Шлифоване

# Етимология

## Шлифовам, шлайфам

- От немски schliff, schleif – изрязвам, шлайфам

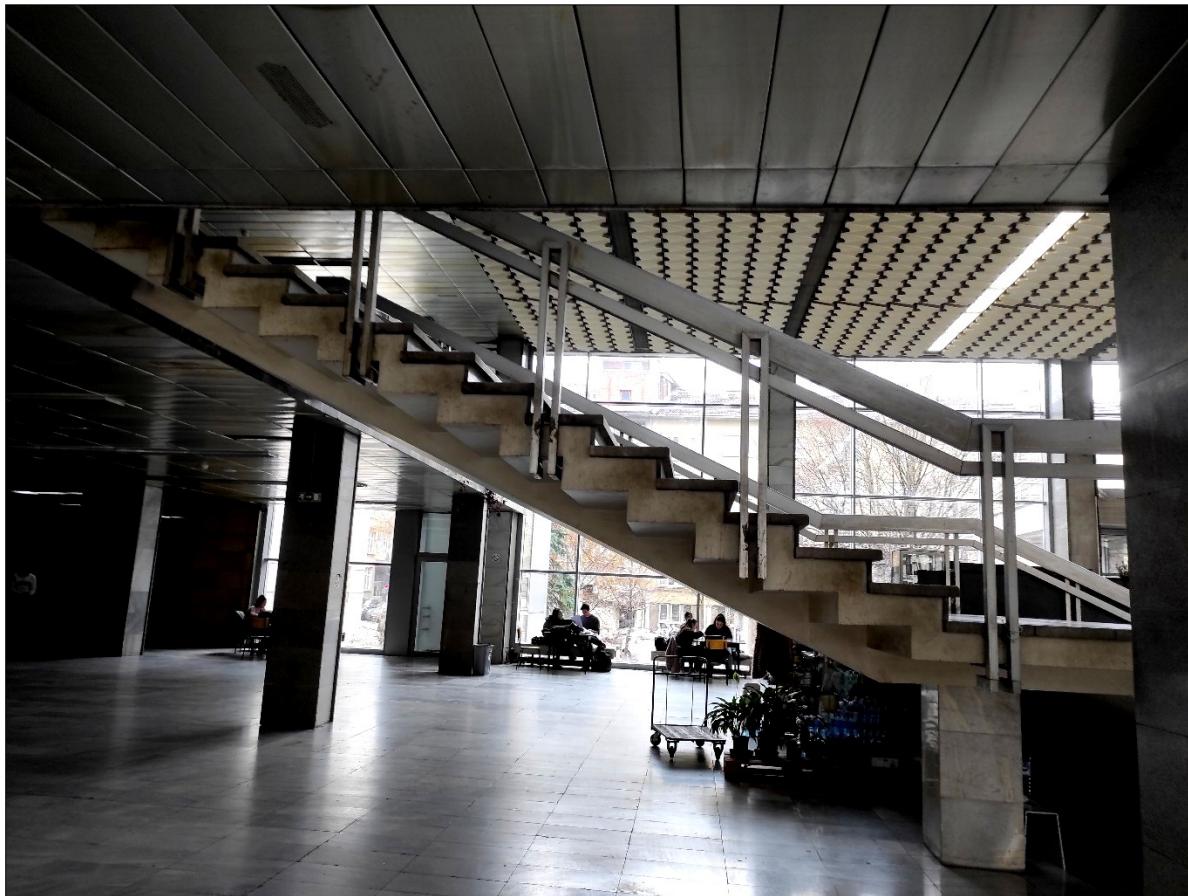
## Обаче

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

# В компютърната графика

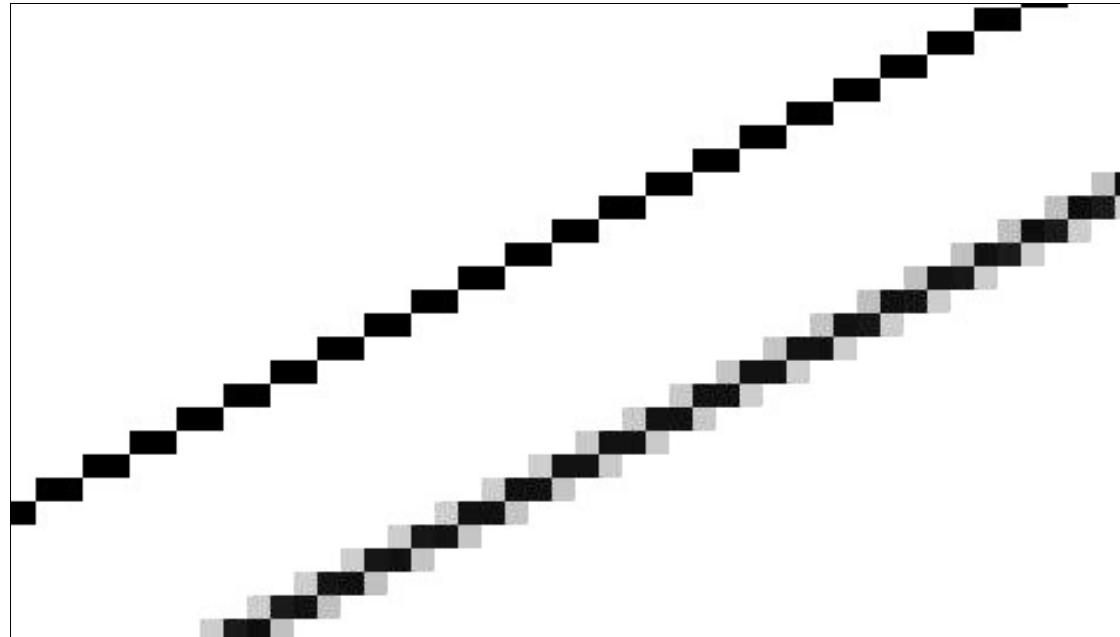
## Проблеми в КГ

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



Снимка: Янина Банкова

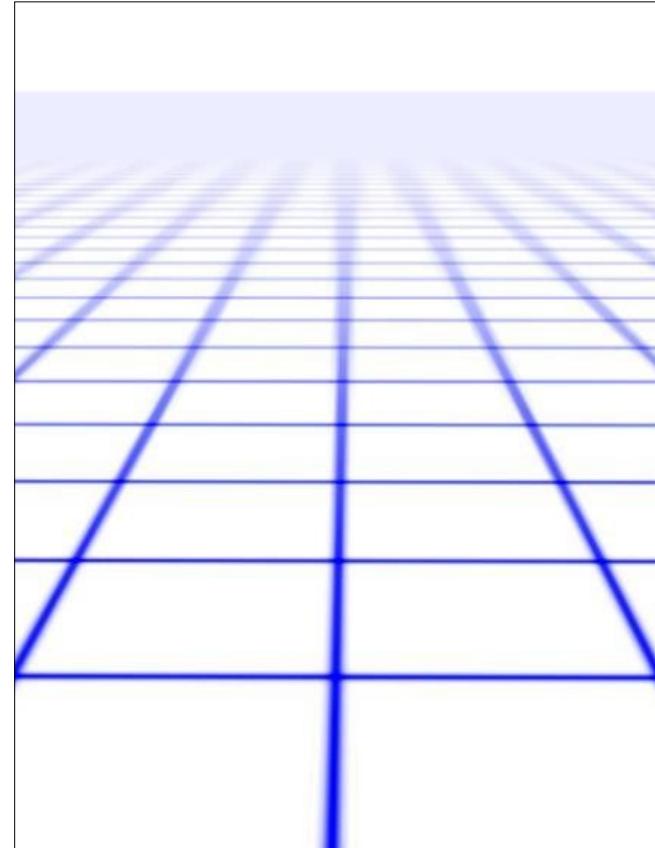
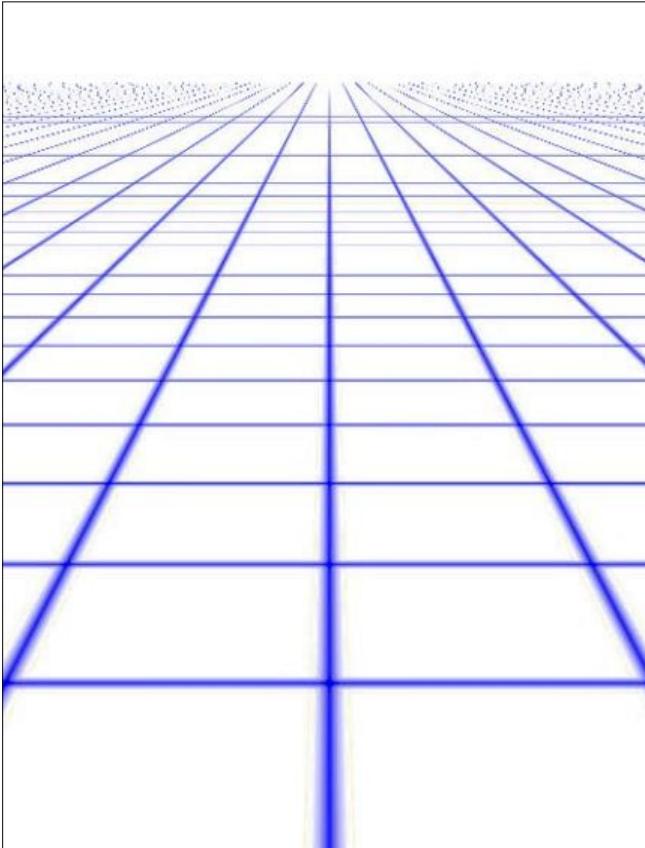
# Изглаждане при линии



# Изглаждане при символи

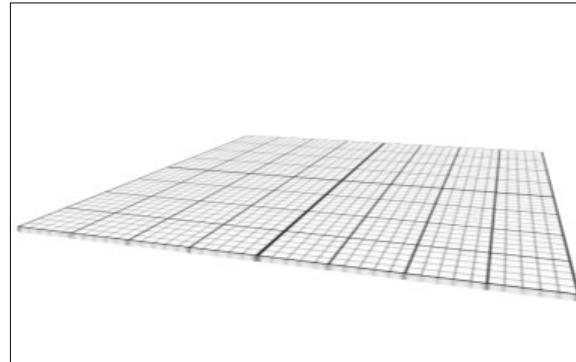


# Заглаждане при текстури



# На живо

- Демонстрация на текстура без и с изглаждане



# Как се изглажда

## Алгоритми за изглаждане

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

## В тази лекция

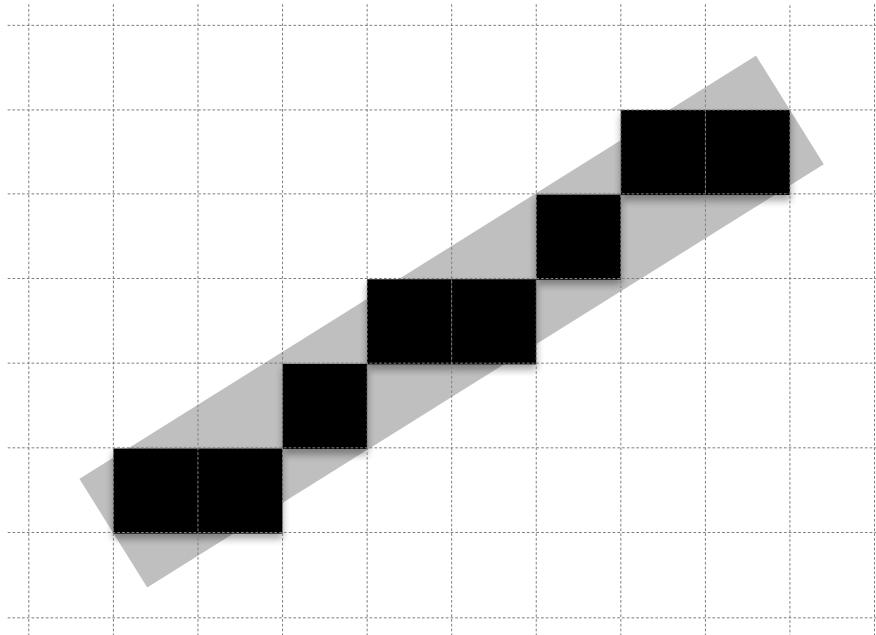
- Ще разгледаме първия алгоритъм  
(останалите вижте в допълнителната литература (най-вече в [LUKI]))

# Подпикселно ниво

## Основна идея

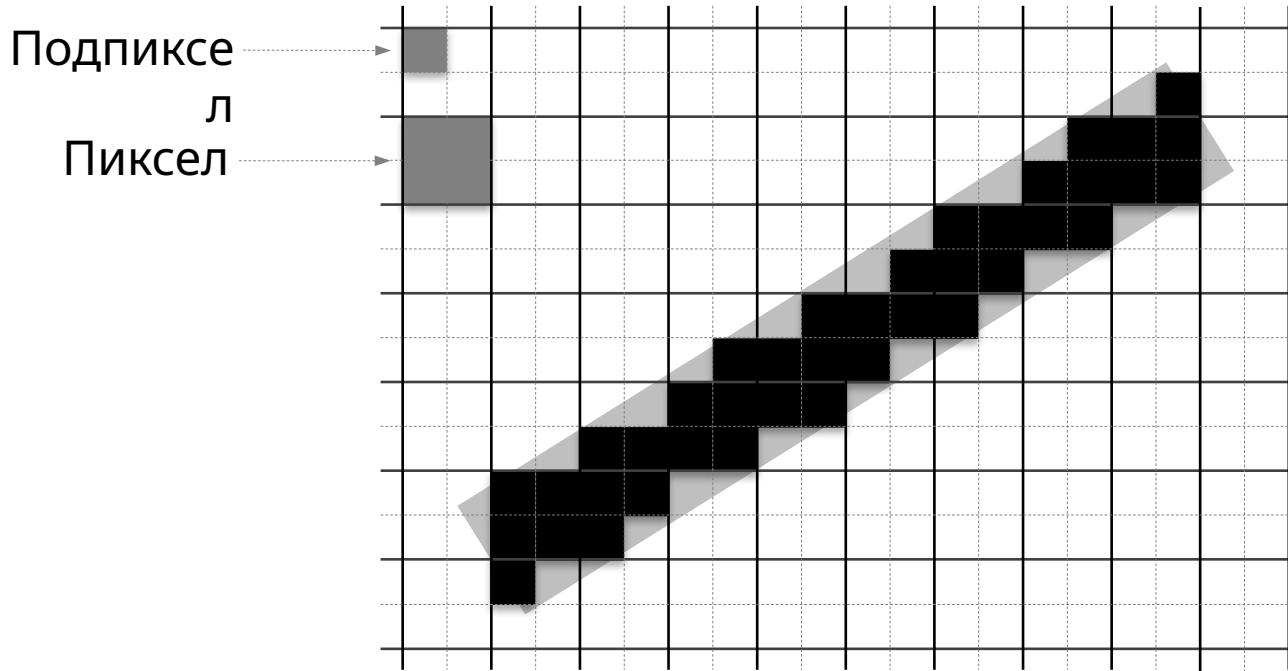
- Всеки пиксел се разбива на подпиксели (subpixels)
- Растеризирането се извършва с подпикселите
- Интензитетът на оригиналните пиксели се определя от броя бели и черни подпиксели

# Растеризация на ниво пиксели

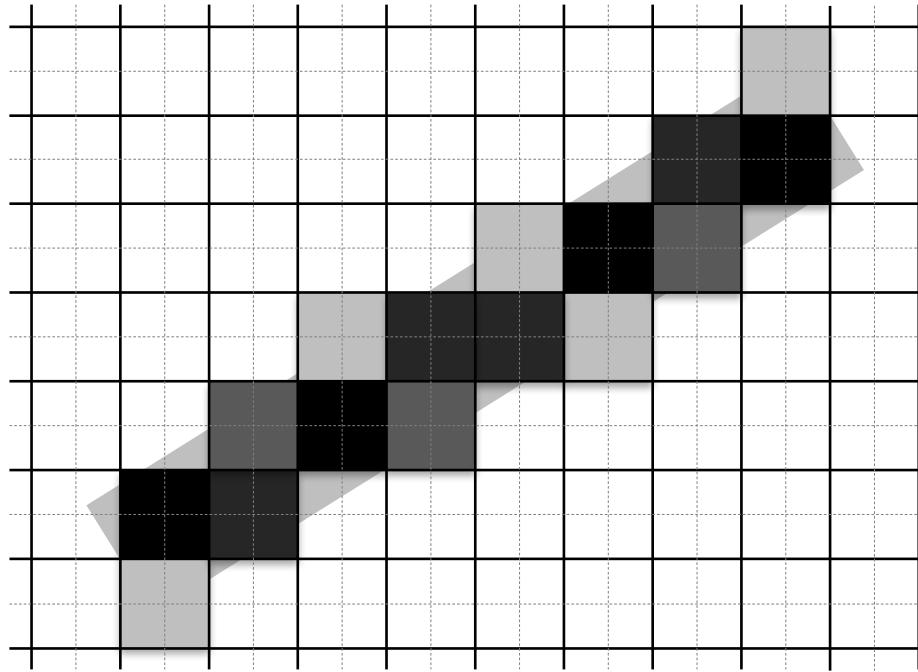
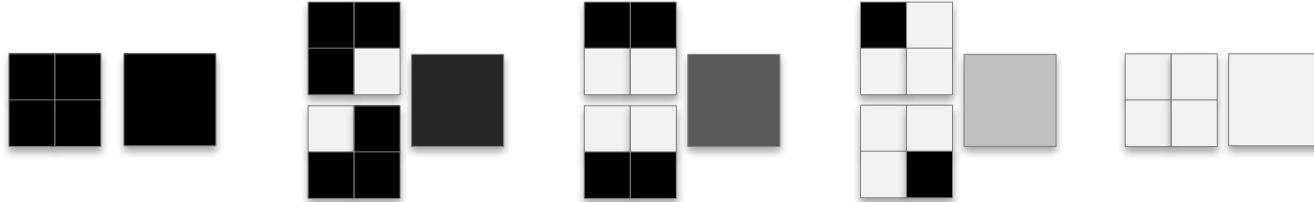


# Растерилизация на ниво подпиксели

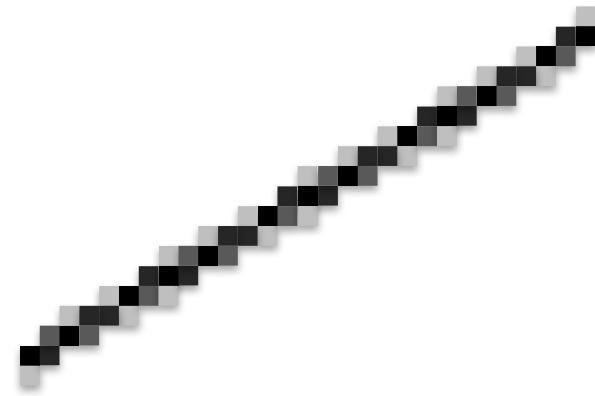
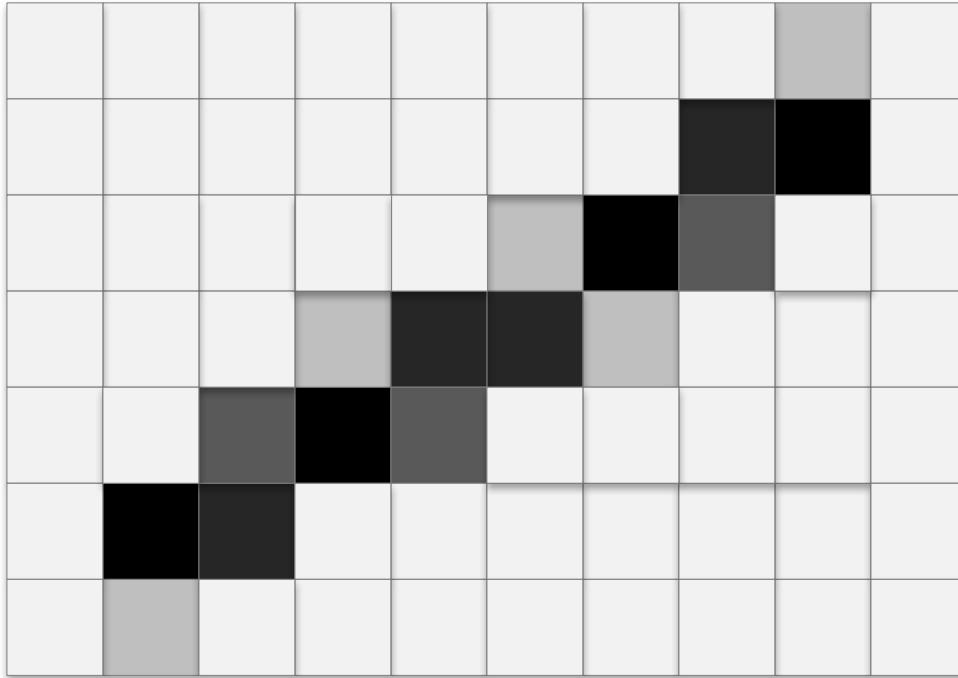
- Растериизира се линия с дебелина
- Един пиксел =  $2 \times 2$  подпиксела



# На ниво пиксели - 5 степени на сивост

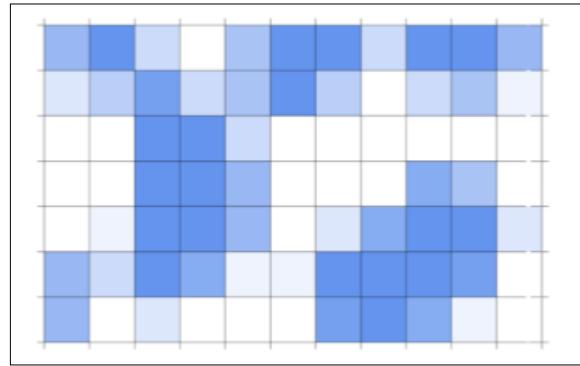


# Краен резултат



# 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>

- Anti-Aliased Line Drawing

<http://courses.engr.illinois.edu/ece390/archive/archive-f2000/mp/mp4/anti.html>

Край