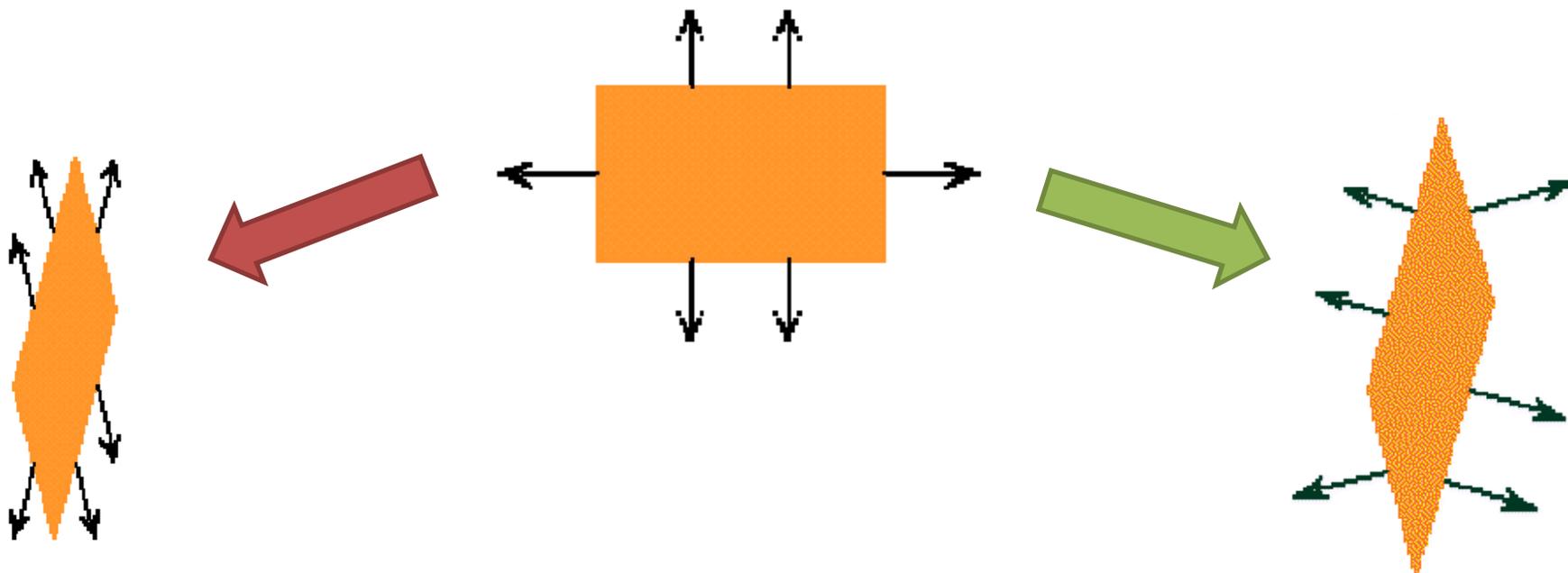


Компьютерная графика

Преобразование нормалей

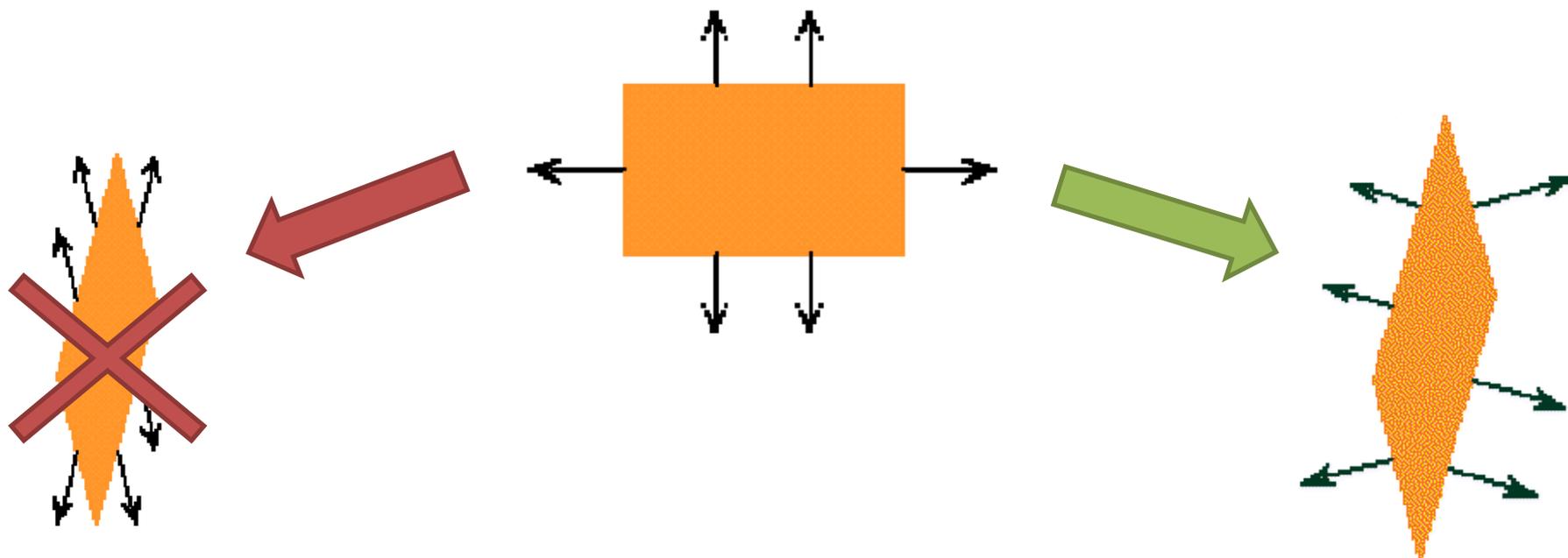
Преобразование нормалей

Прямое преобразование нормалей,
аналогичное тому, как преобразовываются
вектора, **некорректное!**



Преобразование нормалей

Прямое преобразование нормалей, аналогичное тому, как преобразовываются вектора, **некорректное!**



Преобразование нормалей

- Еще раз! Не как точки!
- Преобразование \mathbf{M}
- Преобразование точки $p = (x \ y \ z)$
$$p' = \mathbf{M}p$$
- Нормаль $n = (n_x \ n_y \ n_z)$
- Нельзя $n' = \mathbf{M}n$
- Эта формула не работает для произвольного преобразования

Преобразование нормалей

$$ax + by + cz + d = 0$$

$$\mathbf{n} = (a \quad b \quad c \quad d)^T$$

$$\mathbf{p} = (x \quad y \quad z \quad 1)^T$$

$$\mathbf{n} \cdot \mathbf{p} = \mathbf{n}^T \mathbf{p} = 0$$

$(a \quad b \quad c)$ – нормаль плоскости

d – расстояние от начала СК

Преобразование нормалей

$$0 = \mathbf{n}^T \mathbf{I} \mathbf{p} = \mathbf{n}^T (\mathbf{M}^{-1} \mathbf{M}) \mathbf{p} = (\mathbf{n}^T \mathbf{M}^{-1}) (\mathbf{M} \mathbf{p}) = \mathbf{n}'^T \mathbf{p}'$$

$$\mathbf{p}' = \mathbf{M} \mathbf{p}$$

$$\mathbf{n}' = (\mathbf{n}^T \mathbf{M}^{-1})^T = (\mathbf{M}^{-1})^T \mathbf{n}$$

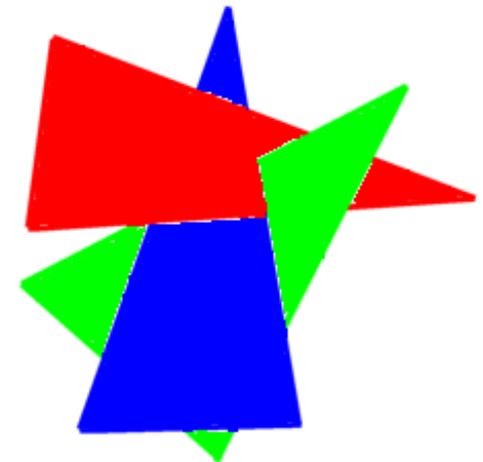
1. Для преобразований общего вида (например, включающих перспективу) надо использовать формулу с верным значением d
2. Для аффинных преобразований d не играет роли, можно положить равным 0
3. Для вращений (обратная и транспонированная матрица равна исходной матрице) можно преобразовывать как точки.

Компьютерная графика

Сортировка граней

Порядок отрисовки граней

- Сортировка художника (прямая сортировка)
- Разделяющие плоскости ($p * n > 0$? 1 : 0)
- Объемлющие сферы = расстояние до центра сферы (чей центр дальше, то рисуется первым)
- Пересечение объектов!!!



Алгоритм Z-буфера

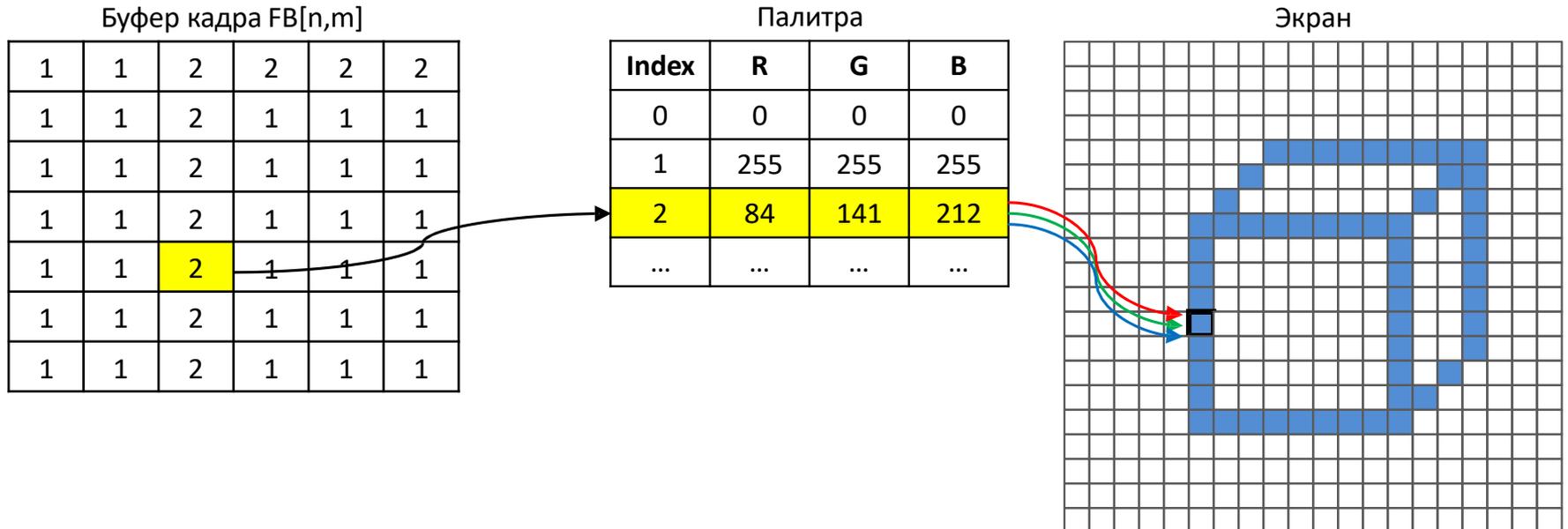
- У каждого пикселя (fragmentShader) есть дополнительная компонента – глубина Z
- Если у текущего пикселя глубина меньше, то он записывается в буфер, иначе – игнорируется.

Компьютерная графика

Дизеринг

Построение палитры

Палитра (LUT)



Разрядность палитры:

1 бит – черно/белый рисунок

4 бита – фиксированная палитра 16 цветов

8 бит – произвольная палитра 256 цветов



Halftoning (dithering)

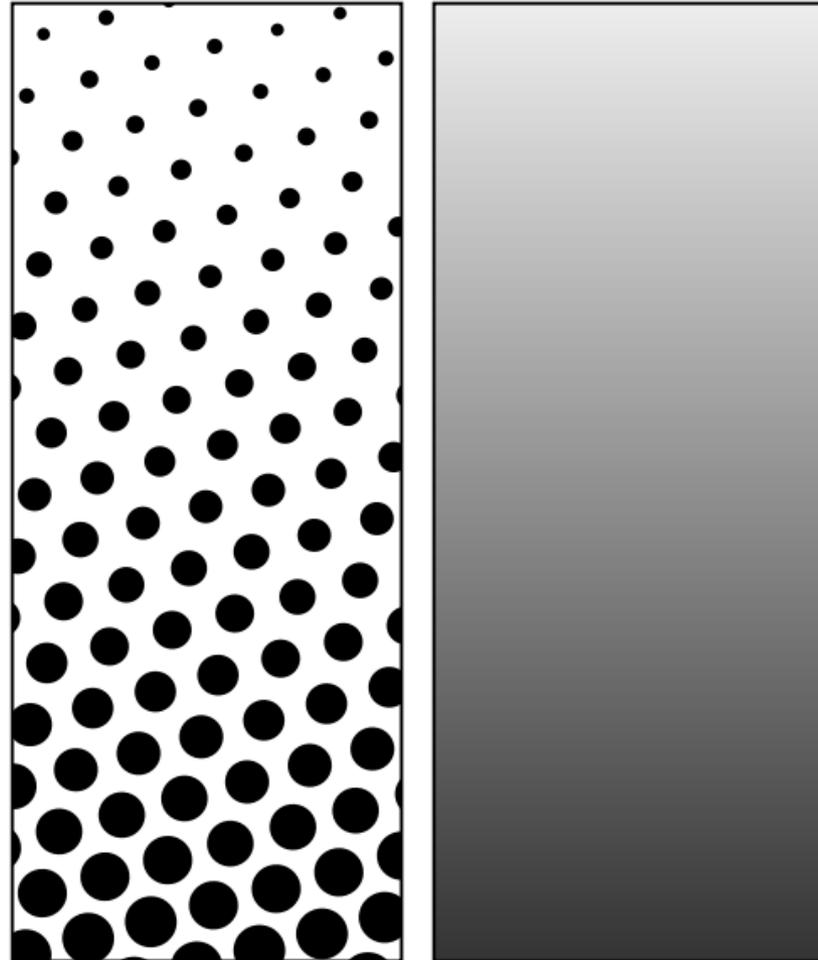
Аппроксимация полутонов

Газетная печать достигается за счет варьирования пространственной (площадной) яркости (черноты).

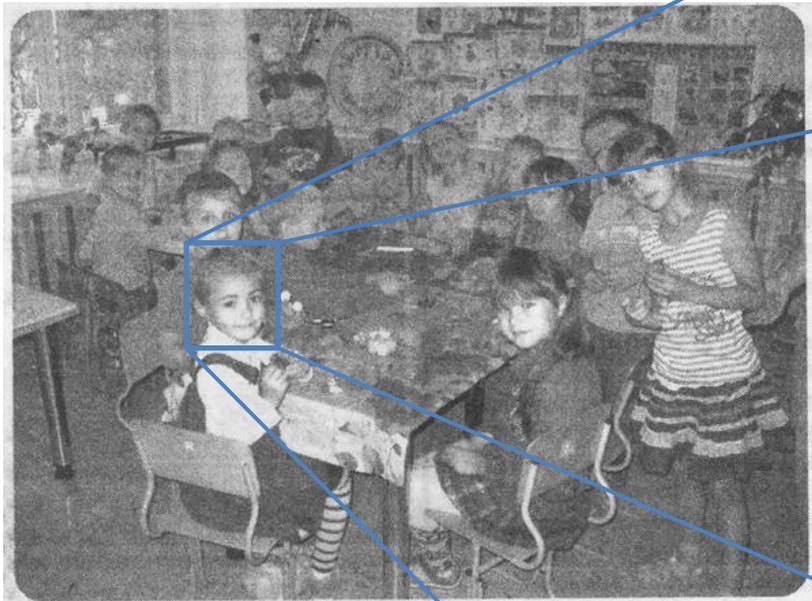
- Газета (офсет): ~85 lpi
- Лазерный принтер (600 dpi): ~100 lpi
- Журнал (офсет): 100–200 lpi

Точки разного диаметра.

Повышение цветового разрешения достигается за счет высокого пространственного разрешения.

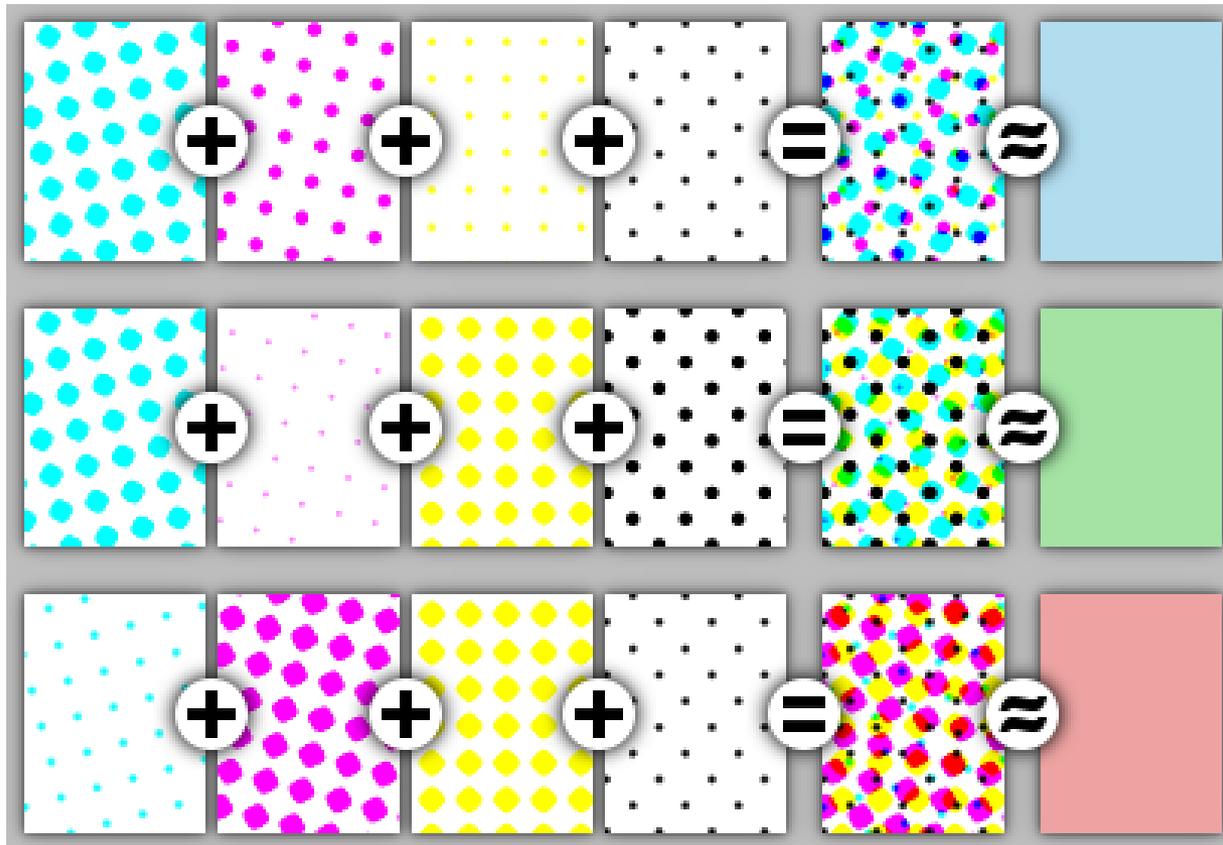


Halftoning: пример



«Навигатор», 21 сентября 2012 г.

В цвете



Зачем нужен дизеринг?

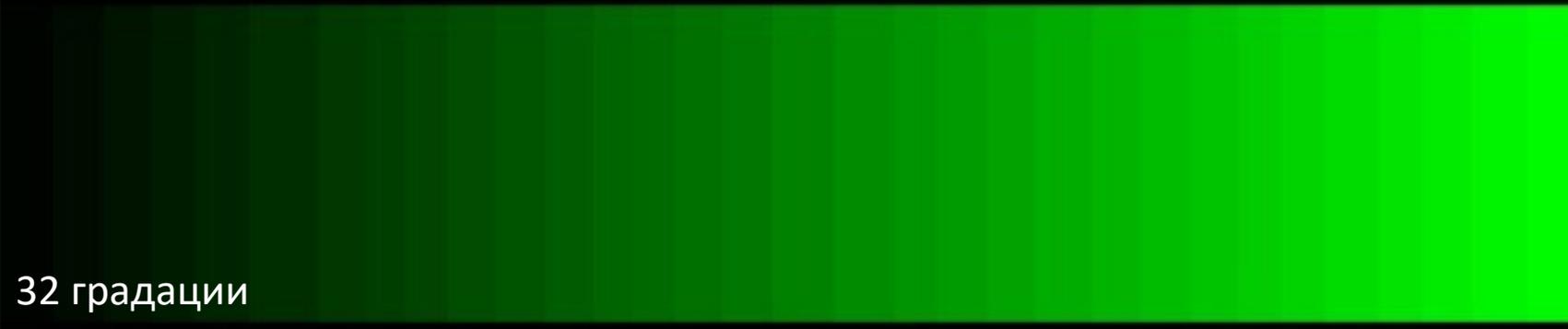
- Печать на принтерах
- Ограничения, накладываемые устройством вывода (дисплей, видеокарта)
- Ограничения, накладываемые форматом файла
- Ограничение памяти, используемой для хранения изображения

Зачем нужен дизеринг?

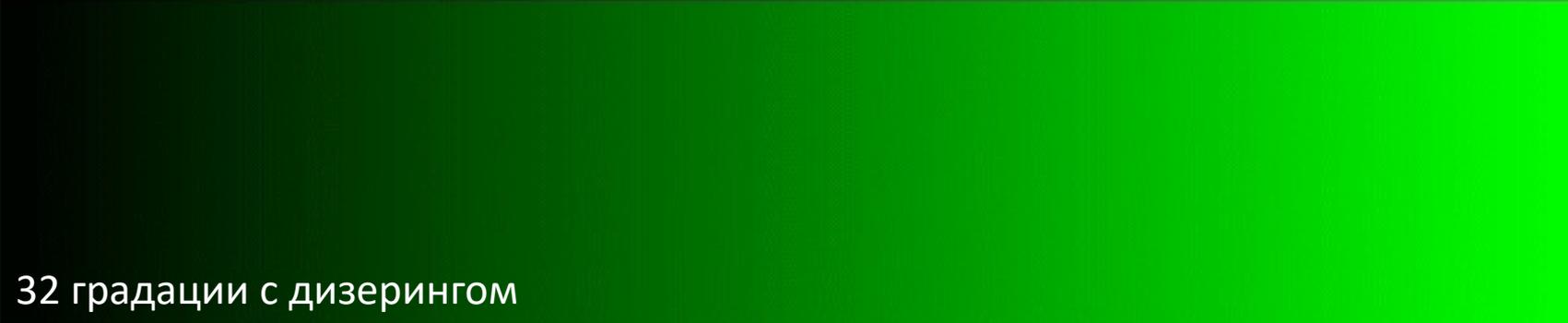
256 градаций



32 градации



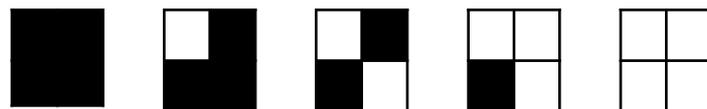
32 градации с дизерингом



Дизеринг с увеличением разрешения

Глаз осуществляет усреднение по пространству, т.е. надо создать шаблоны, которые дают требуемые усредненные яркости.

Например, 5 разных уровней яркости:

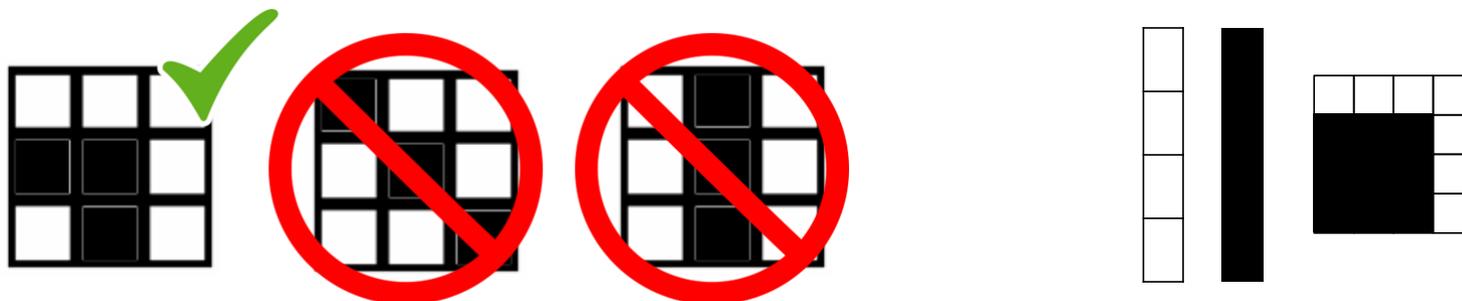


$$2 \times 2 + 1 = 5$$

Дизеринг с увеличением разрешения

Правила построения шаблонов:

- Пиксель, появившийся на одной яркости, должен присутствовать и на более высоких яркостях
- Надо избегать надоедающих артефактов, проявляющихся в области постоянной яркости (см. пример 3x3 – наклонные или вертикальные штрихи)
- Удаление изолированных пикселей для ряда устройств типа лазерных принтеров



Дизеринг с увеличением разрешения

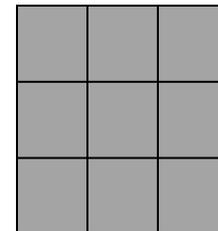
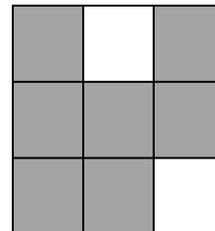
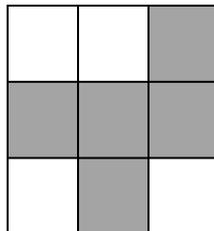
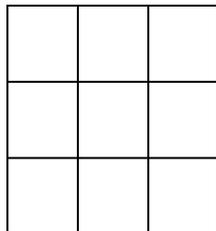
- Матрица дизеринга при использовании повышенного пространственного разрешения

6 8 4

1 0 3

5 2 7

- В целом используя шаблон $K \times K$ можно получить $K \times K + 1$ уровень.



Дизеринг без увеличения разрешения

Пересчет изображения с N интенсивностями без изменения разрешения в палитру, состоящую из K интенсивностей:

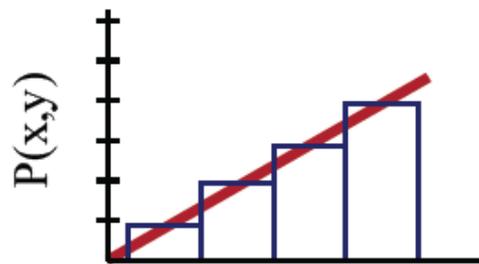
- $K = 2$, только 1 и 0
- $K < N$

Идея дизеринга – сохранение «энергетически» исходного изображения (отклонения от истинных значений в одном пикселе компенсируются коррекцией значений соседних пикселей)

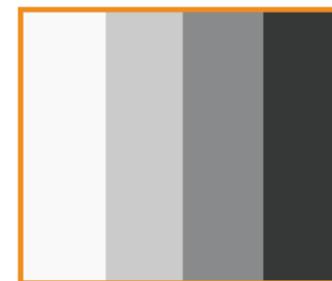
Равномерное квантование



$I(x,y)$



$I(x,y)$



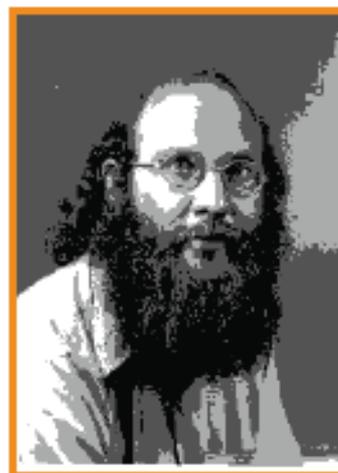
$P(x,y)$



8 бит



4 бита

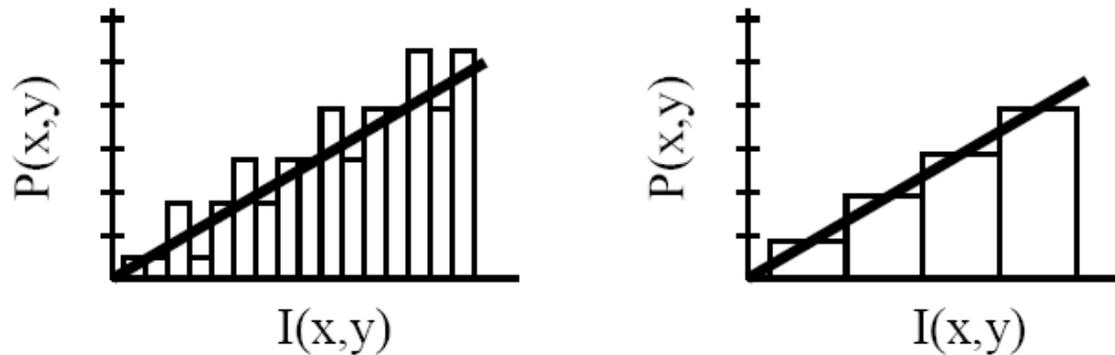


2 бита



1 бита

Дизеринг с зашумлением



$$P(x, y) = \text{trunc}(I(x, y) + \text{noise}(x,y) + 0.5)$$



Оригинал



Uniform quantization



Rand(80)

Дизеринг с шумлением



Uniform quantization



Rand(10)



Rand(20)



Rand(40)



Rand(80)



Rand(160)

Упорядоченный дизеринг

Дизеринг с использованием матрицы, но без увеличения разрешения
Обычно используют матрицы 2×2 , 4×4 , 8×8 , 16×16

$$\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 32 & 8 & 40 & 2 & 34 & 10 & 42 \\ 48 & 16 & 56 & 24 & 50 & 18 & 58 & 26 \\ 12 & 44 & 4 & 36 & 14 & 46 & 6 & 38 \\ 60 & 28 & 52 & 20 & 62 & 30 & 54 & 22 \\ 3 & 35 & 11 & 43 & 1 & 33 & 9 & 41 \\ 51 & 19 & 59 & 27 & 49 & 17 & 57 & 25 \\ 15 & 47 & 7 & 39 & 13 & 45 & 5 & 37 \\ 63 & 31 & 55 & 23 & 61 & 29 & 53 & 21 \end{pmatrix}$$

Упорядоченный дизеринг

$$D_{2n} = \begin{bmatrix} 4D_n & 4D_n + 2U_n \\ 4D_n + 3U_n & 4D_n + U_n \end{bmatrix}$$

$$\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

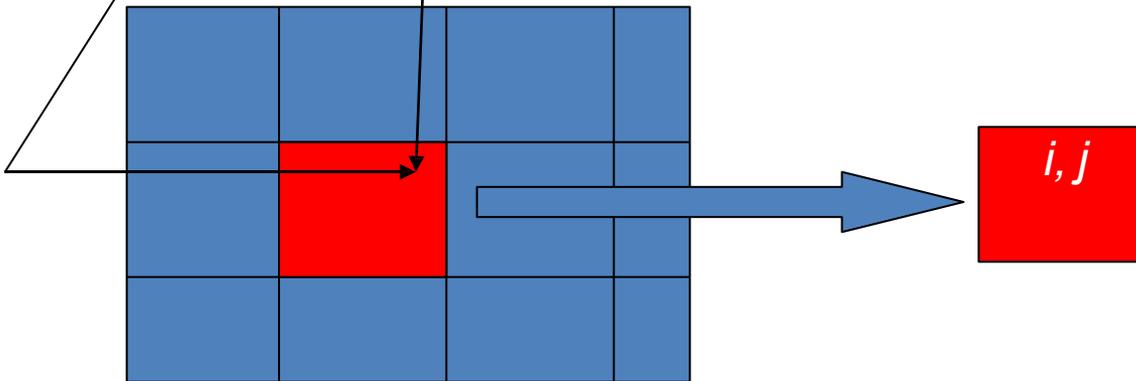
$$\begin{pmatrix} 0 & 32 & 8 & 40 & 2 & 34 & 10 & 42 \\ 48 & 16 & 56 & 24 & 50 & 18 & 58 & 26 \\ 12 & 44 & 4 & 36 & 14 & 46 & 6 & 38 \\ 60 & 28 & 52 & 20 & 62 & 30 & 54 & 22 \\ 3 & 35 & 11 & 43 & 1 & 33 & 9 & 41 \\ 51 & 19 & 59 & 27 & 49 & 17 & 57 & 25 \\ 15 & 47 & 7 & 39 & 13 & 45 & 5 & 37 \\ 63 & 31 & 55 & 23 & 61 & 29 & 53 & 21 \end{pmatrix}$$

Упорядоченный дизеринг

$$i = x \pmod{n}$$

$$j = y \pmod{n}$$

$$I_{\text{out}}(x,y) = \begin{cases} 1, & I_{\text{in}}(x,y) > D_n(i,j) \\ 0, & I_{\text{in}}(x,y) \leq D_n(i,j) \end{cases}$$



0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

Упорядоченный дизеринг



Оригинал



Rand(80)



OD-2



OD-4



OD-8



OD-16

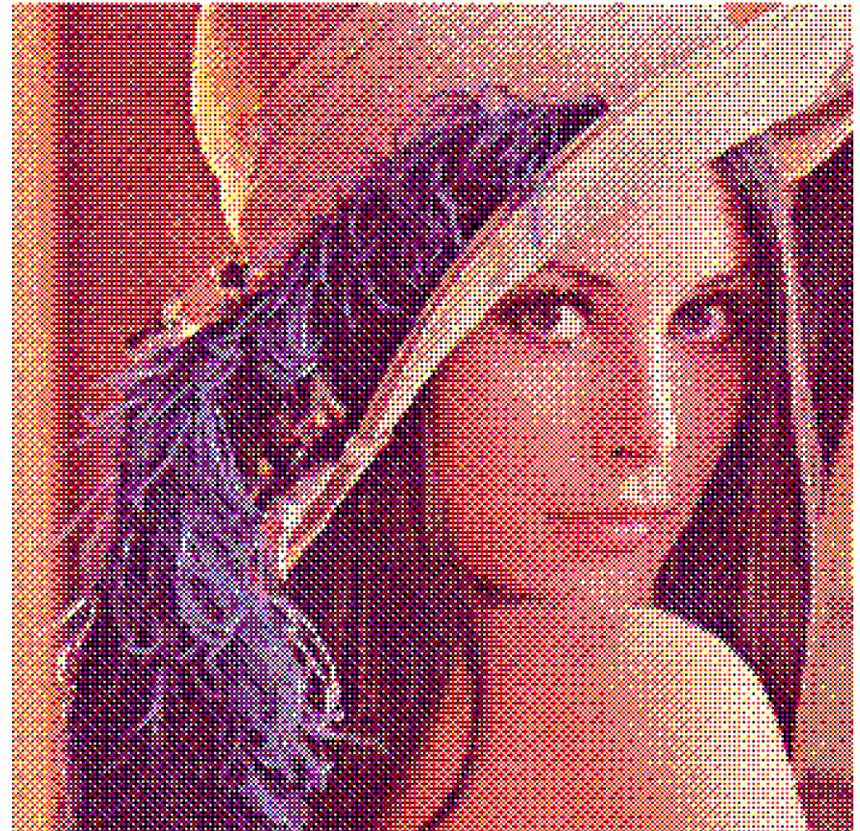
Крупным планом



OD-4



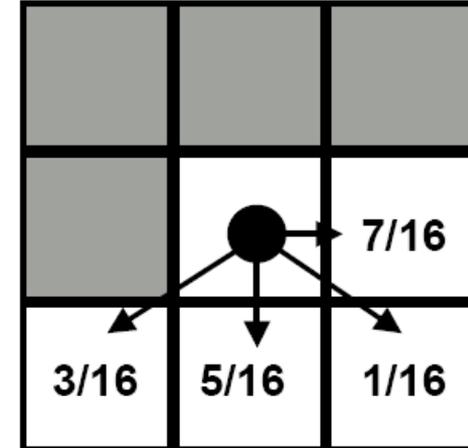
По R, G, B



- Красный – 3 бита,
- Зеленый – 3 бита,
- Синий – 2 бита, т.к. он наименее информативен

Floyd-Steinberg Error Diffusion 1975

- Простая аппроксимация
- Из N градаций в M градаций
- Перебор по строкам растра
- В каждом пикселе рисовать значение, дающее минимальную ошибку (округление)
- Делить ошибку на 4 неравных порции
- Порции ошибки «возвращать» во входное изображение



```
for (y=0; y<height; y++) {  
    for (x=0; x<width; x++) {  
        P(x, y) = trunc(I(x, y) + 0.5);  
        e = I(x, y) - P(x, y); // ошибка  
        I(x+1, y) += 7*e/16;  
        I(x-1, y+1) += 3*e/16;  
        I(x, y+1) += 5*e/16;  
        I(x+1, y+1) += 1*e/16;  
    }  
}
```

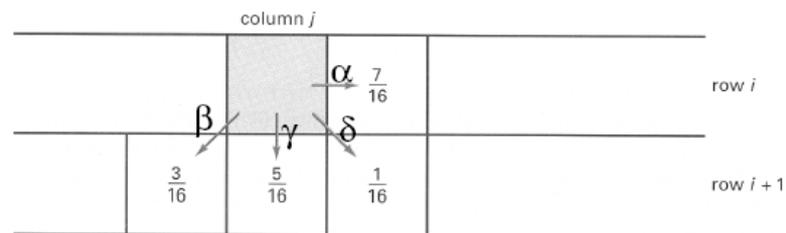
Floyd-Steinberg — результаты



Оригинал



Rand(80)



$$\alpha + \beta + \gamma + \delta = 1.0$$



OD-16



Floyd-Steinberg

Крупным планом



FS

			column j	
				$\alpha \frac{7}{16}$
	β	γ	δ	
	$\frac{3}{16}$	$\frac{5}{16}$	$\frac{1}{16}$	

$$\alpha + \beta + \gamma + \delta = 1.0$$

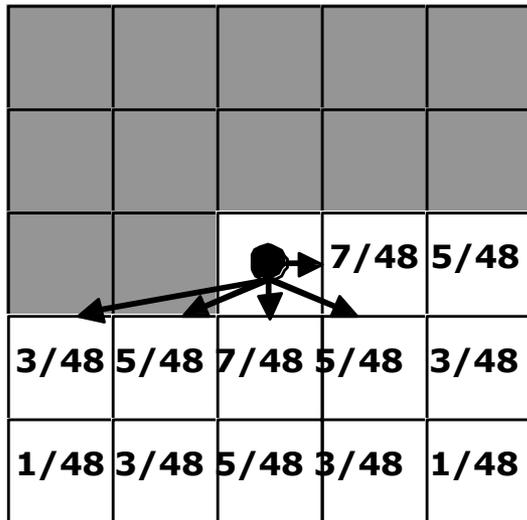
Что делать с границей?



Разброс ошибки по-другому

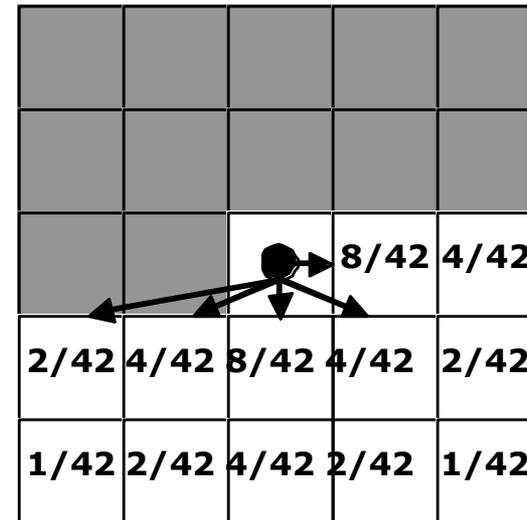
Jarvis dithering, 1976
(Jarvis, Judice, Ninke)

			7/48	5/48
3/48	5/48	7/48	5/48	3/48
1/48	3/48	5/48	3/48	1/48



Stucki dithering, 1981

			8/42	4/42
2/42	4/42	8/42	4/42	2/42
1/42	2/42	4/42	2/42	1/42



Burkes Dithering (1988), Sierra Dithering (1989, несколько) и т. д.

Сравним



Оригинал



По порогу



Ordered



Floyd-Steinberg



Jarvis



Stucki

Тестовые изображения



- Портрет шведской модели Лены Сёдерберг, 1972
- Стандартное тестовое изображение для сравнение алгоритмов обработки изображений

Лена, Lenna

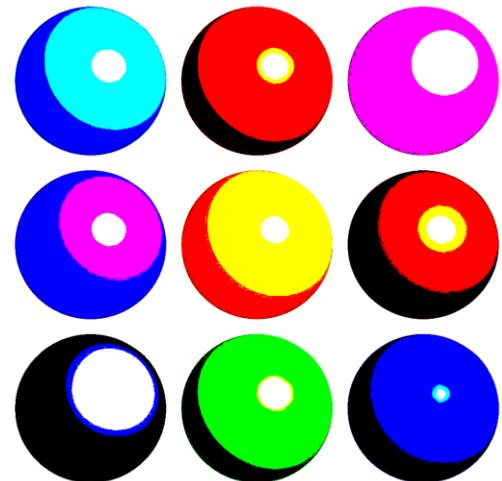
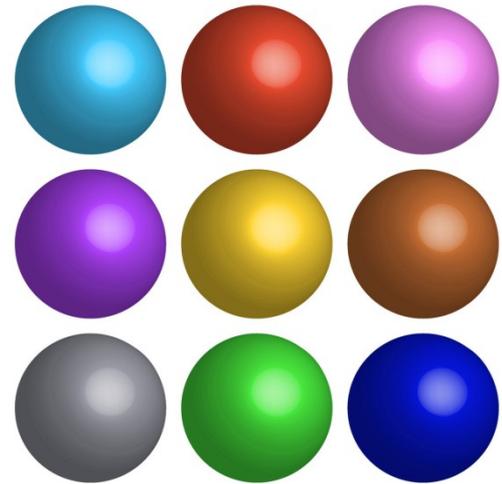
Широкий диапазон цветов,
Наличие резких и плавных границ

Построение палитры

- Цветовой срез
3 бита R, 3 бита G, 2 бита B
(фиксированная палитра)



- Выбор по статистике
- Метод «цветового куба»

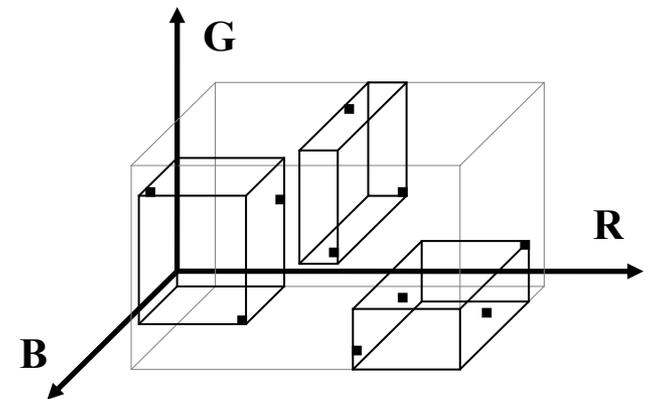
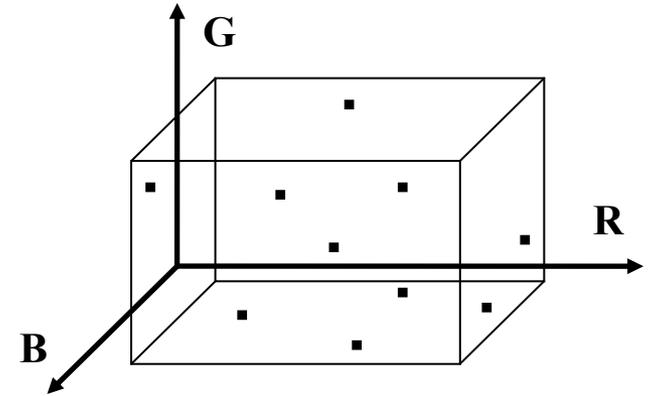


По статистике

- N – полное число пикселей в изображении.
- K – число цветов в палитре.
- Тогда N/K – число пикселей, приходящихся в среднем на один цвет в палитре.
- Находим максимум в статистике, отступаем от него в обе стороны, пока площадь под графиком не превысит N/K – средневзвешенный цвет добавляем в палитру и удаляем данные из статистики.
- **На практике не работает!**

Метод цветового куба

- Строится минимальный параллелепипед, охватывающий все цвета исходной палитры.
- Параллелепипеды итеративно делится на подпараллелепипеды, пока их число не сравняется с размером палитры.
- Цвет в палитре является среднеарифметическим цветом соответствующего подпараллелепипеда.



Метод цветового куба

Выбор подпараллелепипеда: $P = \sum_i w_i \cdot P_i$

• максимальный размер $L_{\max} = \max(L_R, L_G, L_B)$

• число пикселей N

• «ВЫТЯНУТОСТЬ» $\frac{L_{\max}}{L_{\min}}$

