



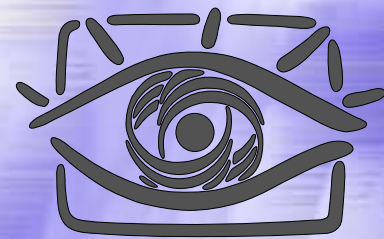
Интерполяция Bayer-pattern

*Из курса лекций «Методы сжатия
медиаданных»*

Дмитрий Ватолин

Video Group

CS MSU Graphics & Media Lab



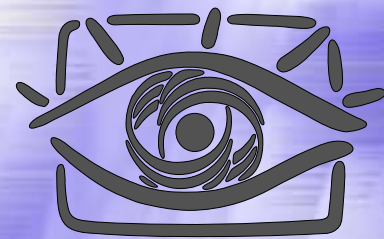
План

- ✓ Устройство матрицы камеры
- ✓ Классификация алгоритмов
 - Не адаптивные
 - Адаптивные
- ✓ Обзор алгоритмов
- ✓ Сравнение алгоритмов
- ✓ Проблемы
- ✓ Результаты

Вид матрицы камеры

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Особенностью матрицы камеры является преобладание зеленых светочувствительных элементов.



GRAPHICS & MEDIA LAB
VIDEO GROUP

Постановка задачи

Преобразовать изображение со светочувствительной матрицы камеры в картинку в формат RGB. Для понимания разности обратимся за помощью к следующему слайду.

Пример

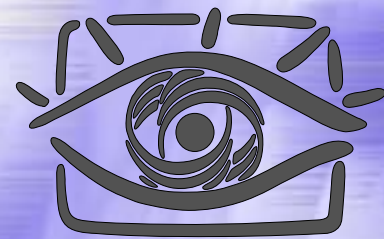
Оригинал



На матрице камеры



Увеличенный вид

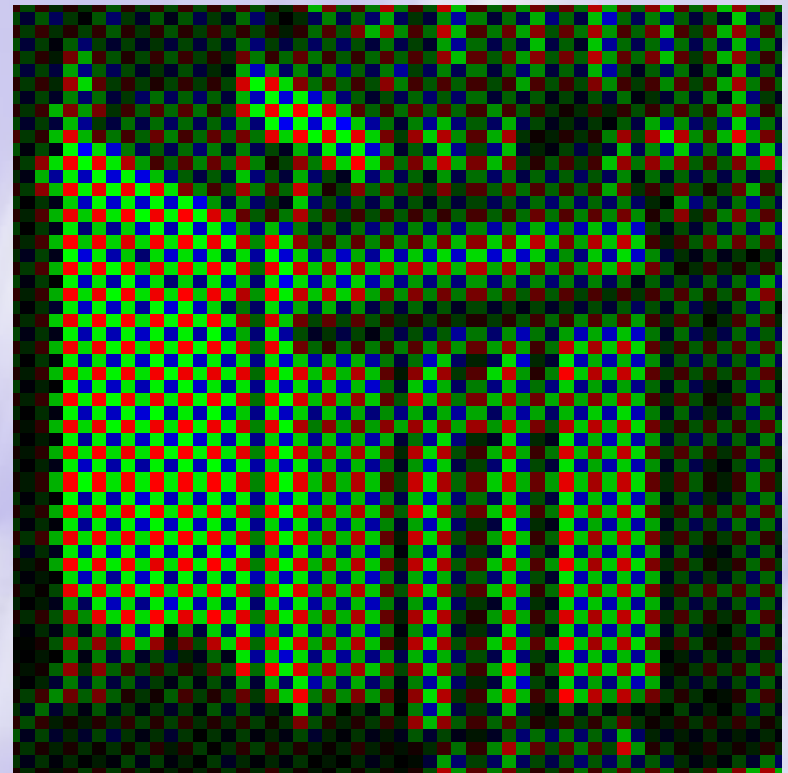


GRAPHICS & MEDIA LAB
VIDEO GROUP

Оригинал



На матрице камеры



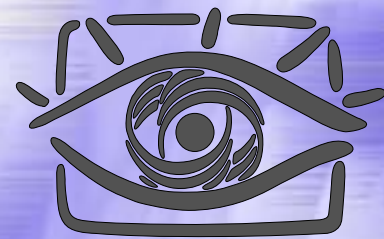
Виды алгоритмов

Не адаптивные

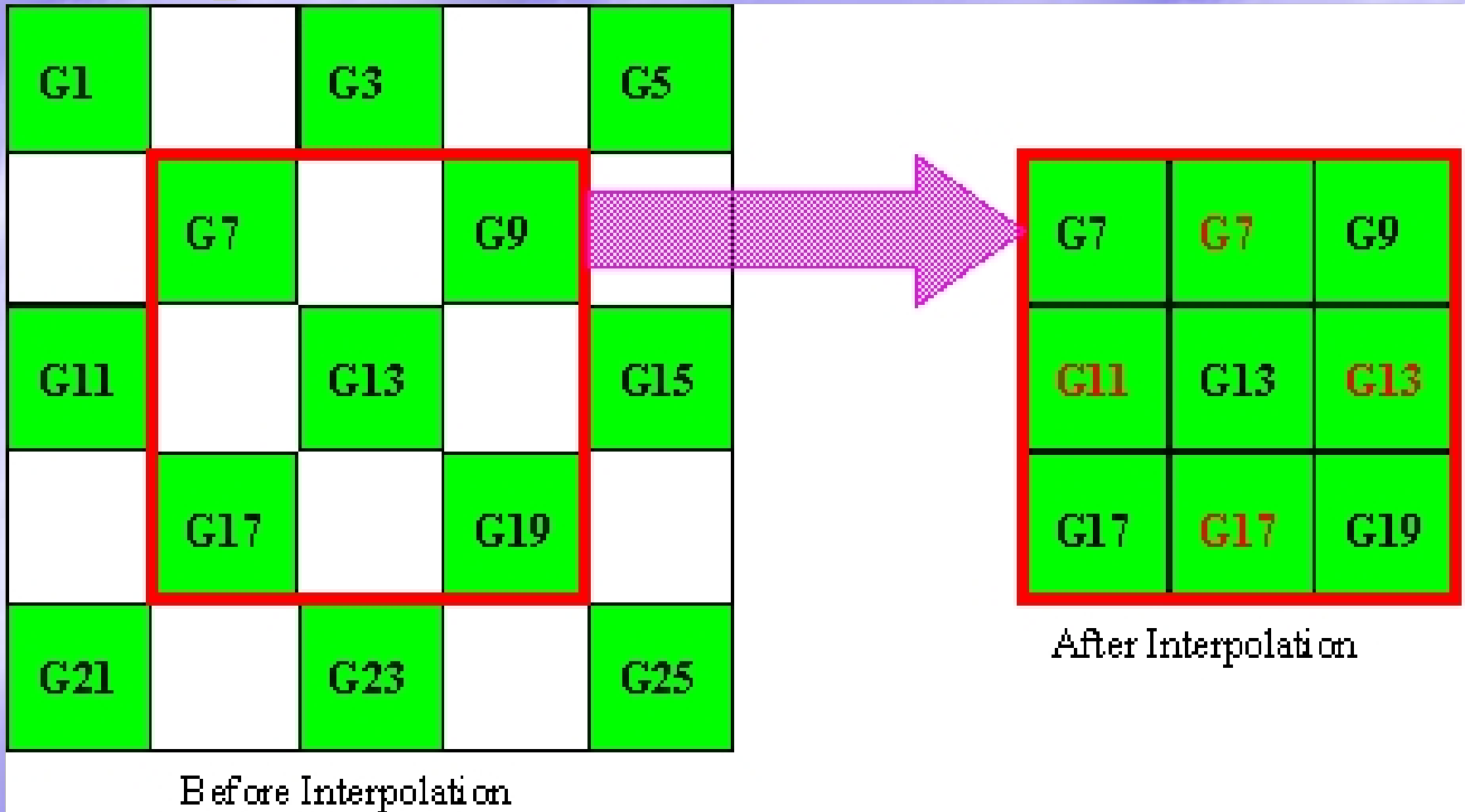
- Nearest Neighbor Replication
- Bilinear Interpolation
- Cubic Convolution Interpolation
- Smooth Hue Transition и .т.п.

Адаптивные

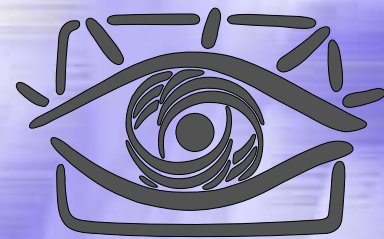
- Interpolation Color Correction
- Edge Sensing Interpolation
- Pattern Recognition Interpolation и т.п.



Nearest Neighbor Replication (NNR)



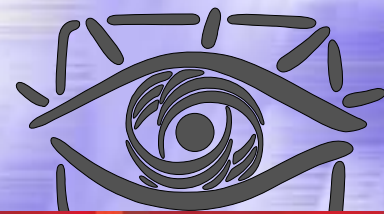
Тестовый пример



GRAPHICS & MEDIA LAB
VIDEO GROUP



Пример NNR

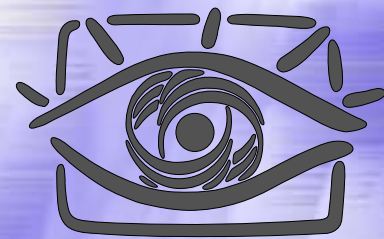


Оригинал



NNR





Bilinear Interpolation

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Green:

$$G8 = (G3 + G7 + G9 + G13) / 4$$

Red/blue в позиции Green:

$$B7 = (B6 + B8) / 2 ;$$

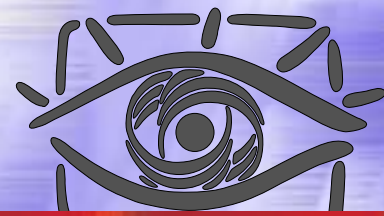
$$R7 = (R2 + R12) / 2$$

Red/blue в позиции blue/red:

$$R8 = (R2 + R4 + R12 + R14) / 4 ;$$

$$B12 = (B6 + B8 + B16 + B18) / 4$$

Пример Bilinear



Оригинал



Bilinear



Cubic Convolution Interpolation (CCI)



Алгоритм описывается так :

$$c(x, y) = \sum_k \sum_l c(x_k, y_l) \cdot h(|x - x_k|) \cdot h(|y - y_l|)$$

,где

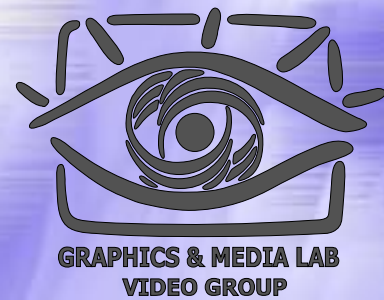
$h(x)$ – ядро;

$c(x_k, y_l)$ – значения функции в узлах;

$c(x, y)$ – интерполированные значения;

Основа алгоритма – определение $h(x)$

Способы определения ядра ССИ



✓ с одним параметром :

$$h(x) = \begin{cases} (a+2) \cdot x^3 - (a+3) \cdot x^2 + 1, & \text{for } 0 \leq x < 1 \\ a \cdot x^3 - 5 \cdot a \cdot x^2 + 8 \cdot a \cdot x + 4 \cdot a, & \text{for } 1 \leq x < 2 \\ 0, & \text{for } 2 \leq x \end{cases}$$

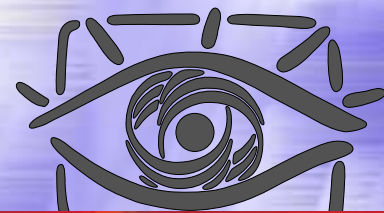
Часто применяемые значения параметров :
 $a - \{-1, 0.5, 0.75\}$

$(b, c) - \{(0.33, 0.33), (1.5, -0.25)\}$

✓ с двумя параметрами :

$$h(x) = \begin{cases} \frac{1}{6} \cdot (-9 \cdot b - 6 \cdot c + 12) \cdot x^3 - \frac{1}{6} \cdot (12 \cdot b + 6 \cdot c - 18) \cdot x + \frac{1}{6} \cdot (-2 \cdot b + 6), & \text{for } 0 \leq x < 1 \\ \frac{1}{6} \cdot (-b - 6 \cdot c) \cdot x^3 - \frac{1}{6} \cdot (6 \cdot b + 30 \cdot c) \cdot x^2 + \frac{1}{6} \cdot (-12 \cdot b - 48 \cdot c) \cdot x + \frac{1}{6} \cdot (8 \cdot b + 24 \cdot c), & \text{for } 1 \leq x < 2 \\ 0, & \text{for } 2 \leq x \end{cases}$$

Пример ССИ



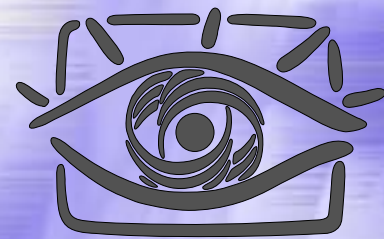
Оригинал



ССИ



Smooth Hue Transition Interpolation



GRAPHICS & MEDIA LAB
VIDEO GROUP

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Green:

$$G8 = (G3 + G7 + G9 + G13) / 4$$

Red/blue в позиции Green:

$$B7 = G7 / 2 * (B6 / G6 + B8 / G8)$$

$$B13 = G13 / 2 * (B8 / G8 + B18 / G18)$$

Red/blue в позиции blue/red:

$$B12 = G12 / 4 * (B6 / G6 + B8 / G8 + B16 / G16 + B18 / G18)$$

Smooth Hue Transition Interpolation (log)



G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Green:

$$G8 = (G3 + G7 + G9 + G13) / 4$$

Преобразование:

$$G \rightarrow G_{\log}, B \rightarrow B_{\log}, R \rightarrow R_{\log}$$

Red/blue в позиции Green:

$$B7_{\log} = G7_{\log} + (B6_{\log} - G6_{\log} + B8_{\log} - G8_{\log}) / 2$$

$$B13_{\log} = G13_{\log} + (B8_{\log} - G8_{\log} + B18_{\log} - G18_{\log}) / 2$$

Red/blue в позиции blue/red:

$$B12_{\log} = G12_{\log} + (B6_{\log} - G6_{\log} + B8_{\log} - G8_{\log} + B16_{\log} - G16_{\log} + B18_{\log} - G18_{\log}) / 4$$

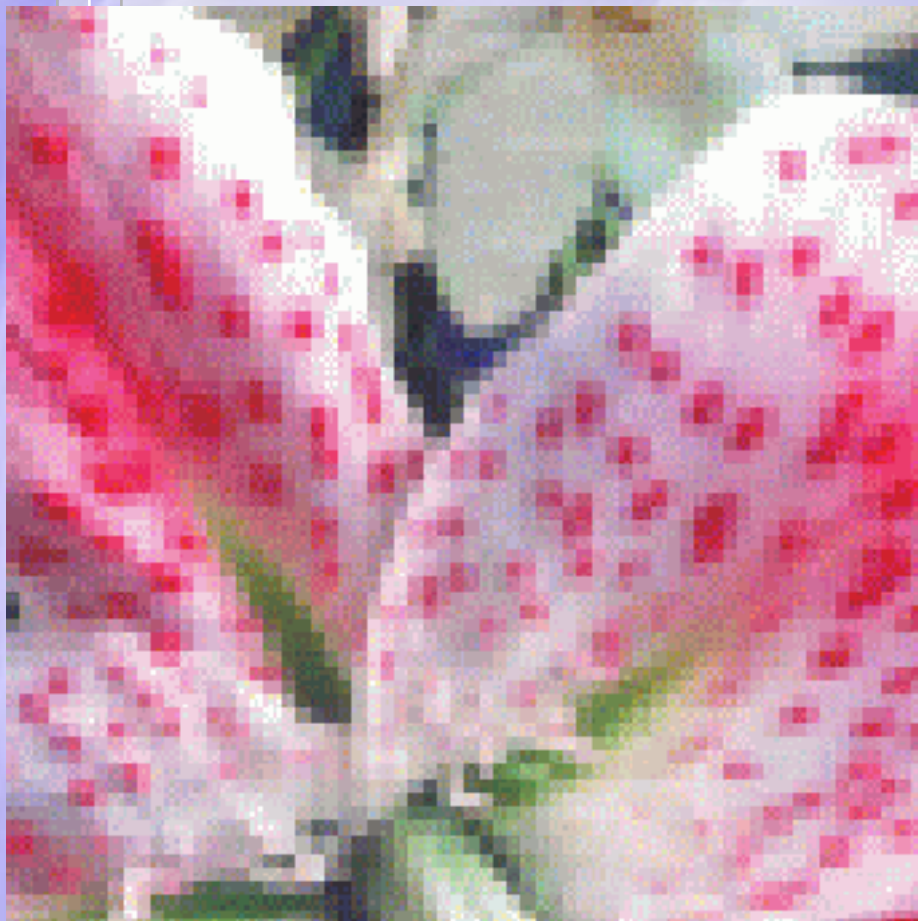
Обратное преобразование:

$$G_{\log} \rightarrow G, B_{\log} \rightarrow B, R_{\log} \rightarrow R$$

Сравнение алгоритмов



Оригинал

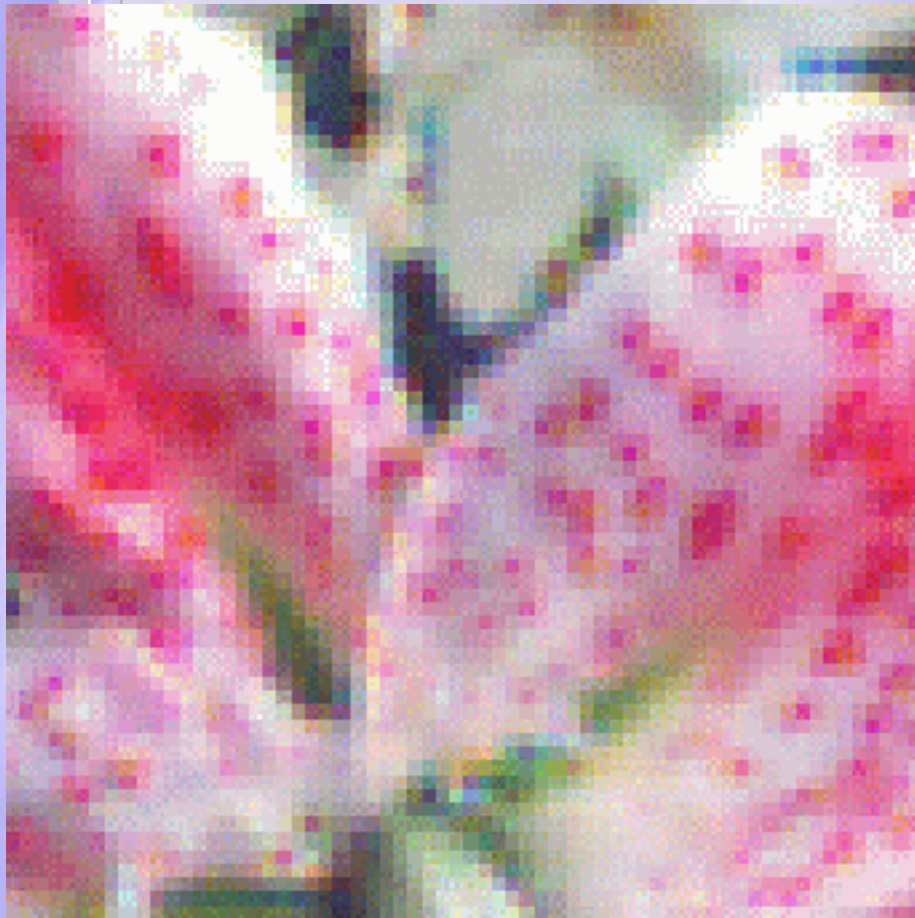


NNR



Сравнение алгоритмов

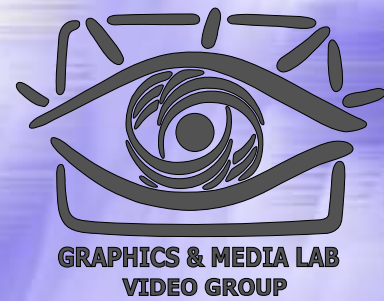
Bilinear



Smooth

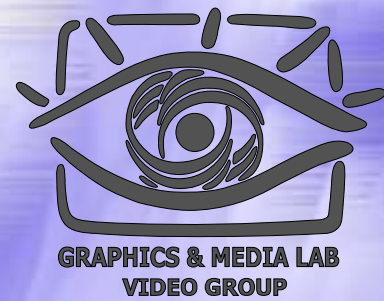


Адаптивные алгоритмы



- Edge Sensing Interpolation
- Interpolation with Color Correction
- Pattern Recognition Interpolation и т.п.

Edge Sensing Interpolation



G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Green:

$$\Delta H = |G7 - G9| \text{ and } \Delta V = |G3 - G13|$$

If $\Delta H < T$ AND $\Delta V < T$,

$$G8 = \frac{(G7 + G9)}{2};$$

Else if $\Delta H > T$ AND $\Delta V < T$,

$$G8 = \frac{(G3 + G13)}{2};$$

Else

$$G8 = \frac{(G3 + G7 + G9 + G13)}{4};$$

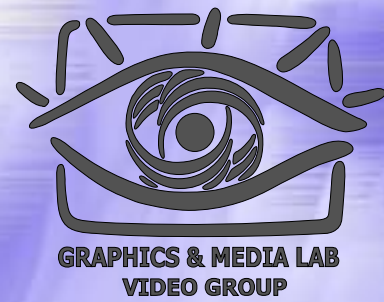
End

где T – порог (задается);

Red/blue:

определяется как в Smooth Hue Transition Interpolation (log)

Interpolation with Color Correction



Green :

$$\Delta H = |G4 - G6| + |B5 - B3 + B5 - B7|$$

$$\Delta V = |G2 - G8| + |B5 - B1 + B5 - B9|$$

If $\Delta H < \Delta V$,

$$G5 = (G4 + G6) / 2 + (B5 - B3 + B5 - B7) / 4;$$

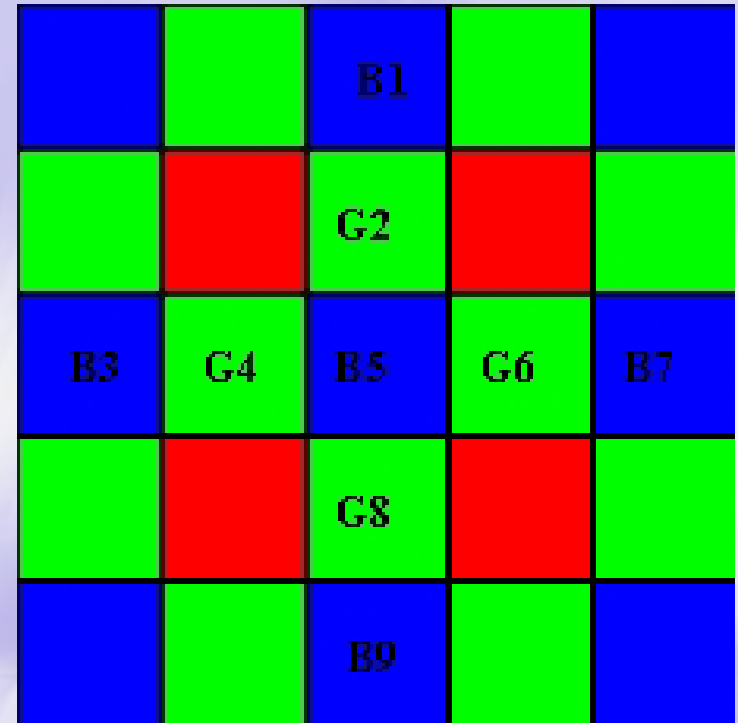
Else if $\Delta H > \Delta V$,

$$G5 = (G2 + G8) / 2 + (B5 - B1 + B5 - B9) / 4;$$

Else

$$G5 = (G2 + G4 + G6 + G8) / 4 + (B5 - B1 + B5 - B3 + B5 - B7 + B5 - B9) / 8;$$

End



Interpolation with Color Correction

Red/blue в позиции Green

$$R4 = (R1 + R7) / 2 + (G4 - G1 + G4 - G7) / 4$$

$$R2 = (R1 + R3) / 2 + (G2 - G1 + G2 - G3) / 4$$

Red/blue в позиции blue/red

$$\Delta N = |R1 - R9| + |G5 - G1 + G5 - G9|$$

$$\Delta P = |R3 - R7| + |G5 - G3 + G5 - G7|$$

If $\Delta N < \Delta P$,

$$R5 = (R1 + R9) / 2 + (G5 - G1 + G5 - G9) / 2;$$

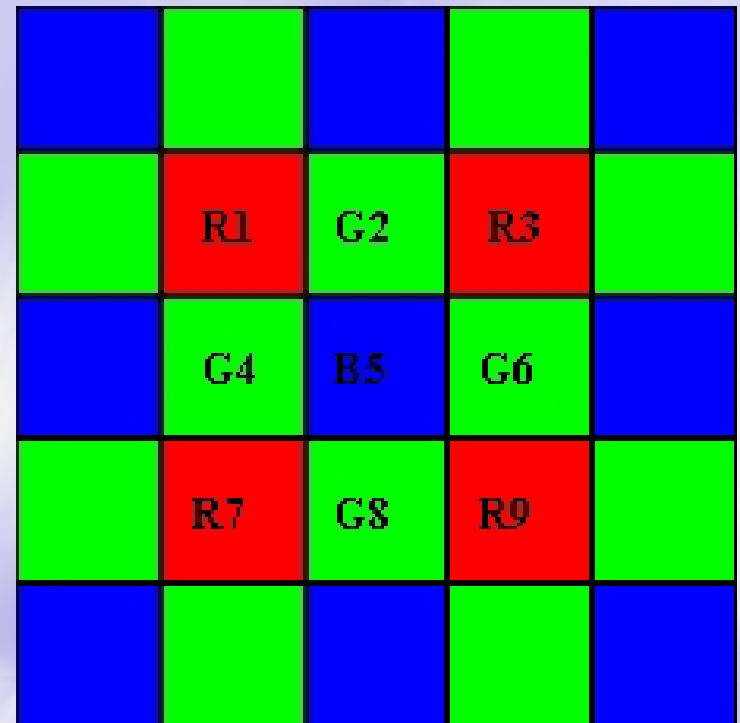
Else if $\Delta N > \Delta P$,

$$R5 = (R3 + R7) / 2 + (G5 - G3 + G5 - G7) / 2;$$

Else

$$R5 = (R1 + R3 + R7 + R9) / 4 + (G5 - G1 + G5 - G3 + G5 - G7 + G5 - G9) / 4;$$

End



Pattern Recognition

Interpolation



G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Green

ИСПОЛЬЗУЮТСЯ ШАБЛОНЫ

H H H H
H ? H H ? H H ? H H ? H
L L L L
(a) (b) (c) (d)

a,d – граница (edge);

b – полоса (*stripe*);

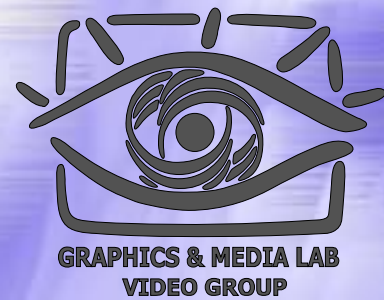
c – угол (corner);

Red/blue

любой ранее описанный
МЕТОД

Pattern Recognition

Interpolation



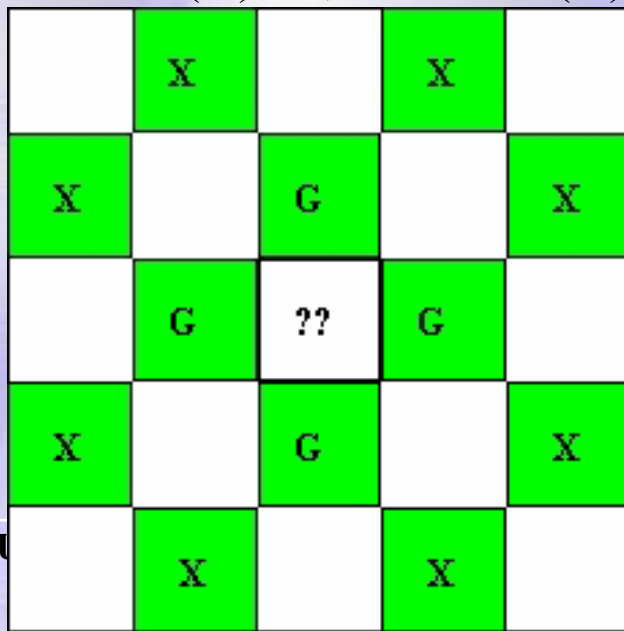
Граница

$$G_{12} = \text{median}\{G_7, G_{11}, G_{13}, G_{17}\} // (A, B, C, D)$$

Полоса

$$G_{??} = \text{clip}_C^B(2M - S)$$

$$M = \text{SUM}(G) / 4, S = \text{SUM}(X) / 8$$

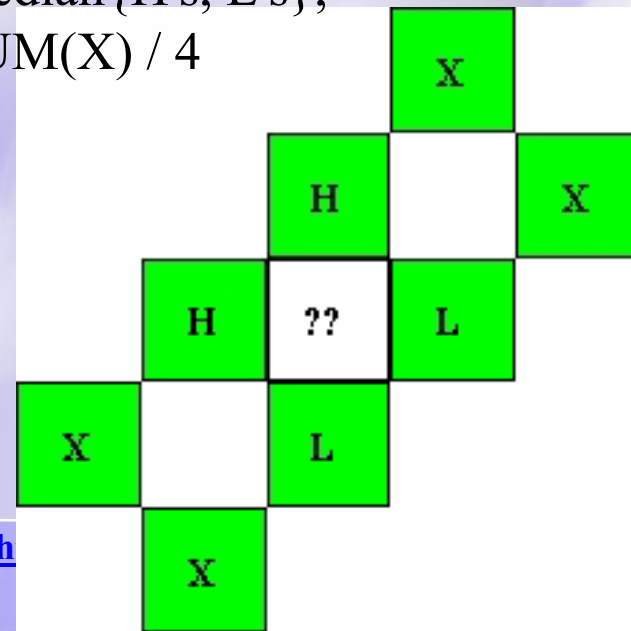


Угол

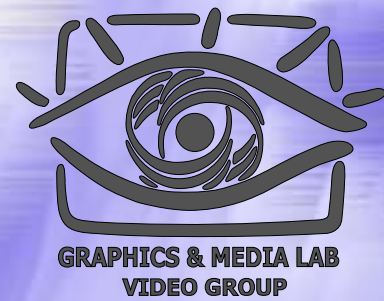
$$G_{??} = \text{clip}_C^B(2M - S)$$

$$M = \text{median}\{H's, L's\},$$

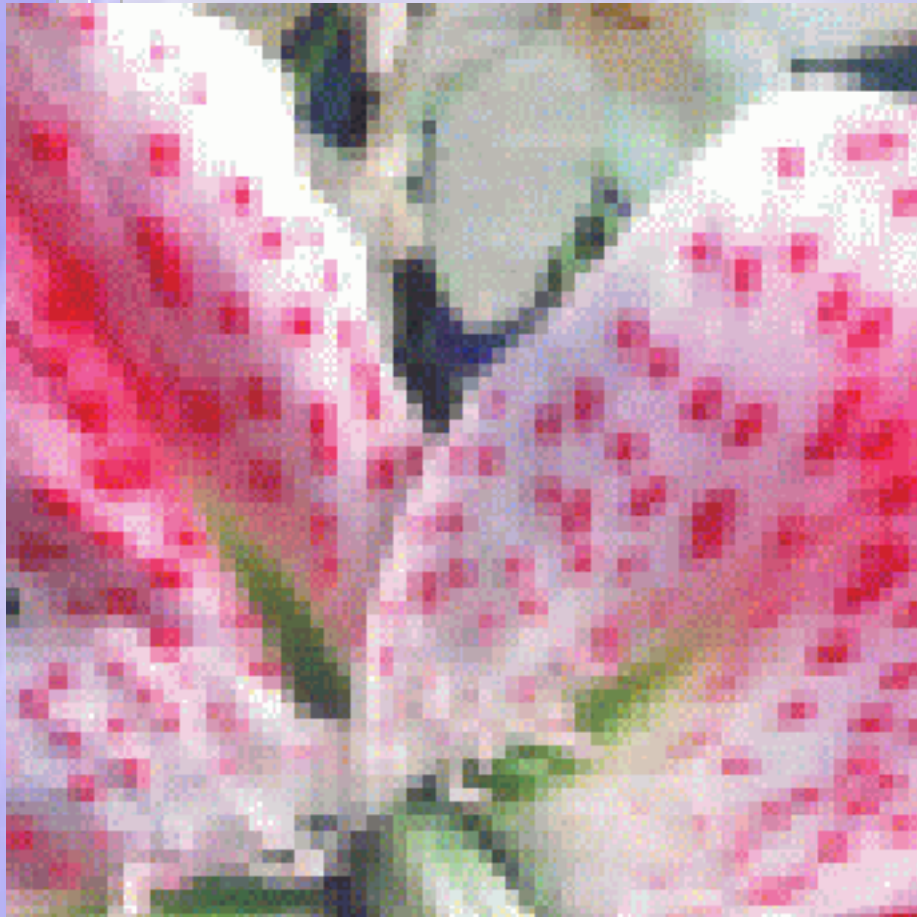
$$S = \text{SUM}(X) / 4$$



Сравнение алгоритмов



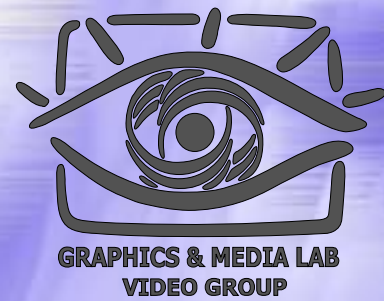
Оригинал



Edge Sensing



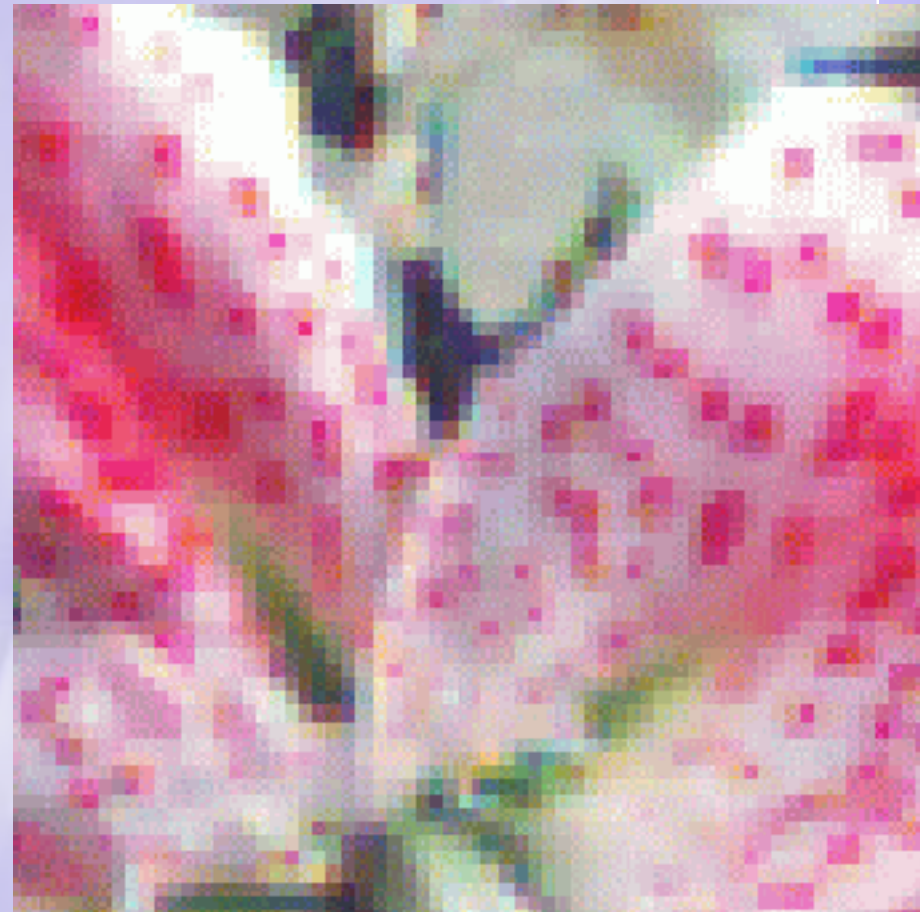
Сравнение алгоритмов



Color Correction



Pattern Recognition



Mean Square Error (MSE)

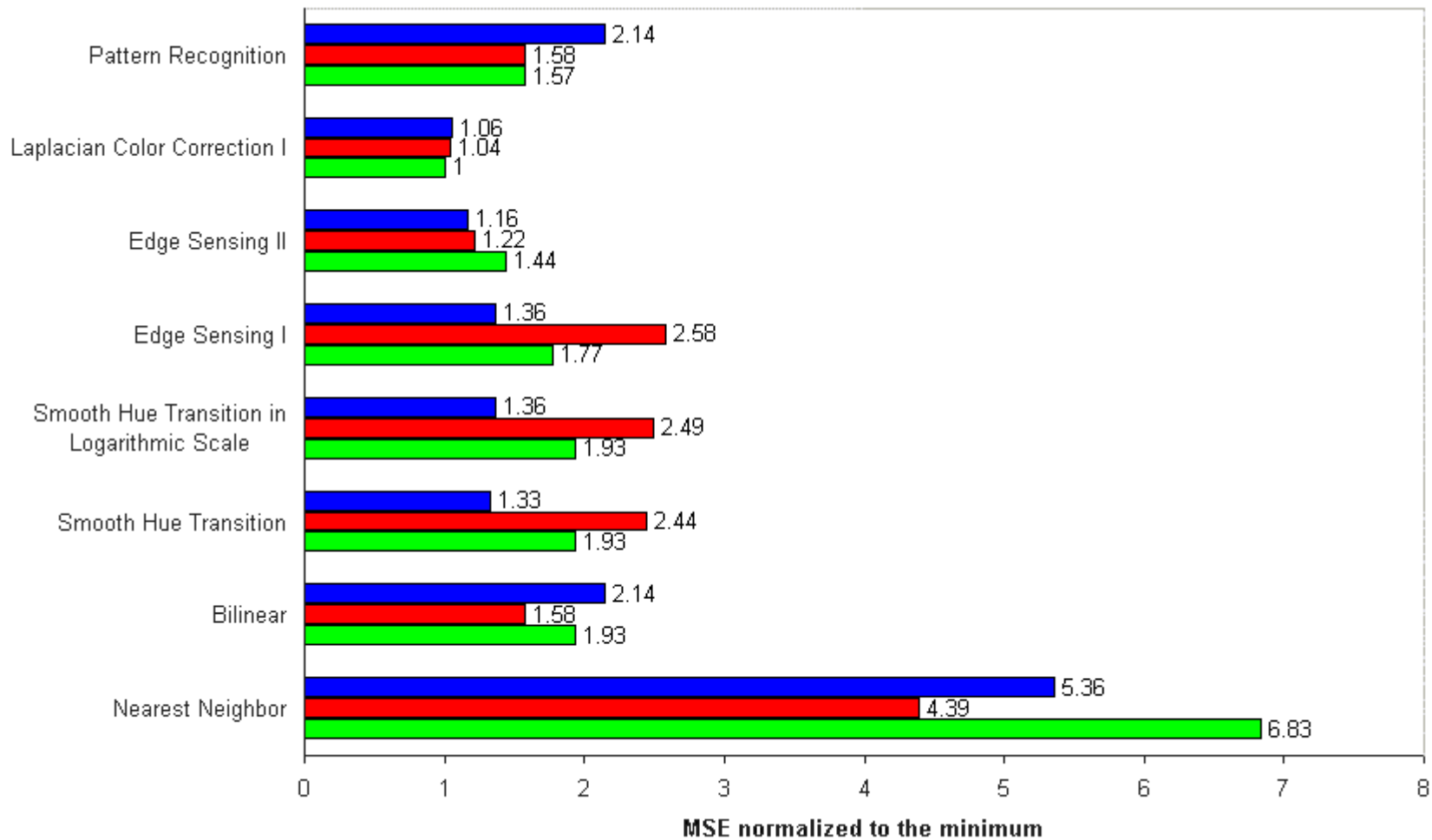
$$MSE = \frac{\sum_{x=1}^m \sum_{y=1}^n [I_0(x,y) - I_r(x,y)]^2}{mn}$$

где

I_0 – оригинал;

I_r – после интерполяции;

Mean Square Error (MSE)

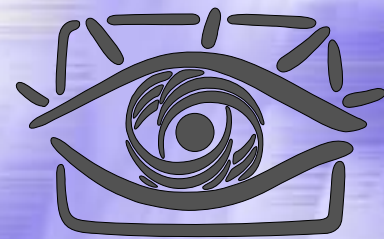


Проблемы

При работе с чёрно-белым изображением возникают артефакты, проявляющиеся в виде цветных пятен.



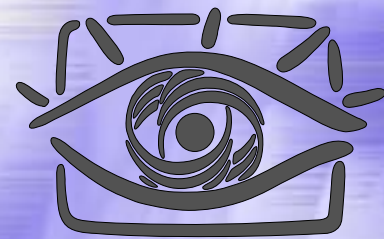
Проблемы



GRAPHICS & MEDIA LAB
VIDEO GROUP



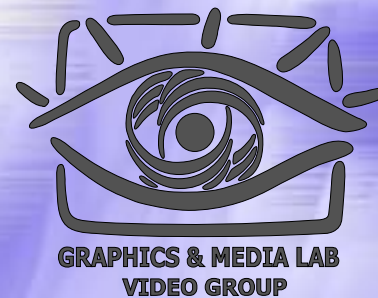
Проблемы



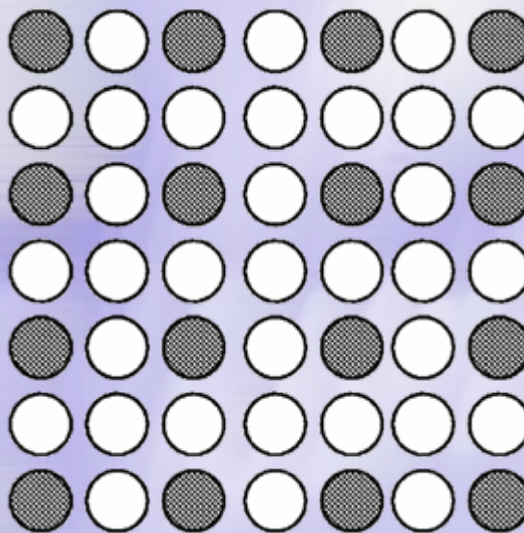
GRAPHICS & MEDIA LAB
VIDEO GROUP



NEW EDGE-DIRECTED INTERPOLATION (NEDI)



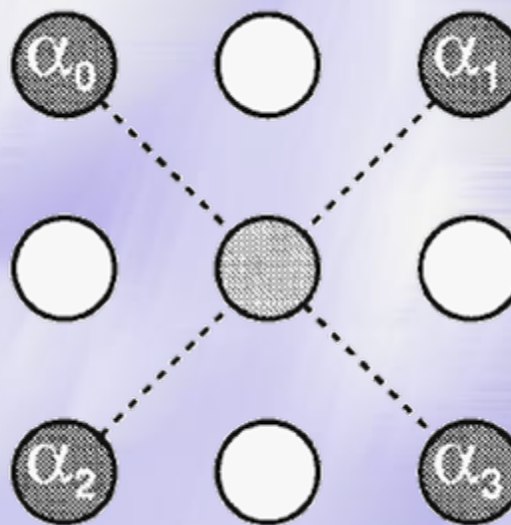
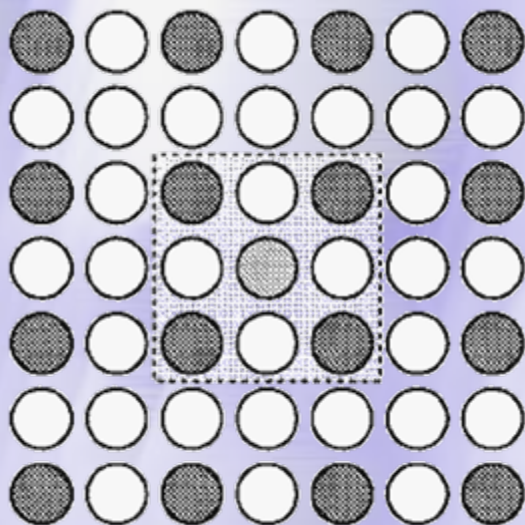
Имеется изображение Y размера $2W \times 2H$, а также изображение X размера $W \times H$ полученное непосредственно из Y

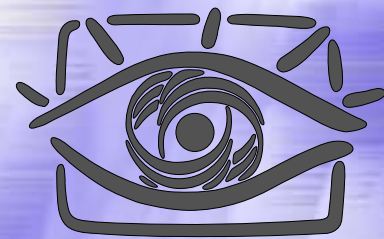


$$Y_{2i,2j} = X_{i,j}$$

NEDI – область действия

Nedi восстанавливает(увеличивает)
уменьшенное изображение используя
интерполяцию по соседним пикселям





NEDI - интерполяция

Интерполяция по четырем соседям:

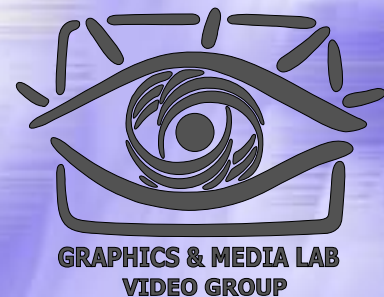
$$\hat{Y}_{2i+1,2j+1} = \sum_{k=0}^1 \sum_{l=0}^1 \alpha_{2k+l} Y_{2(i+k),2(j+l)} \quad \text{ИЛИ}$$

$$F\left(2\vec{x} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}, n\right) = \sum_{j=0}^1 \sum_{i=0}^1 \alpha_{2j+i} F\left(2\vec{x} + \begin{pmatrix} 2i \\ 2j \end{pmatrix}, n\right)$$

Мера близости 2-х изображений:

$$SSE = \sum_{k,l} \left(F(\vec{x}_{k,l}, n) - \sum_{j=0}^1 \sum_{i=0}^1 \alpha_{2j+i} F\left(2\vec{x}_{k,l} + \begin{pmatrix} 2i-1 \\ 2j-1 \end{pmatrix}, n \right) \right)^2$$

NEDI - основная идея

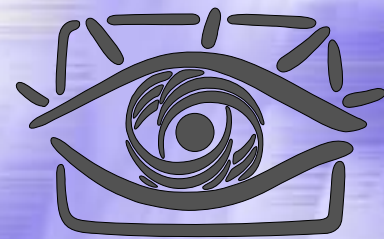


Nedi основывается на геометрической схожести оригинального и уменьшенного изображений

Рассматривается матричное уравнение

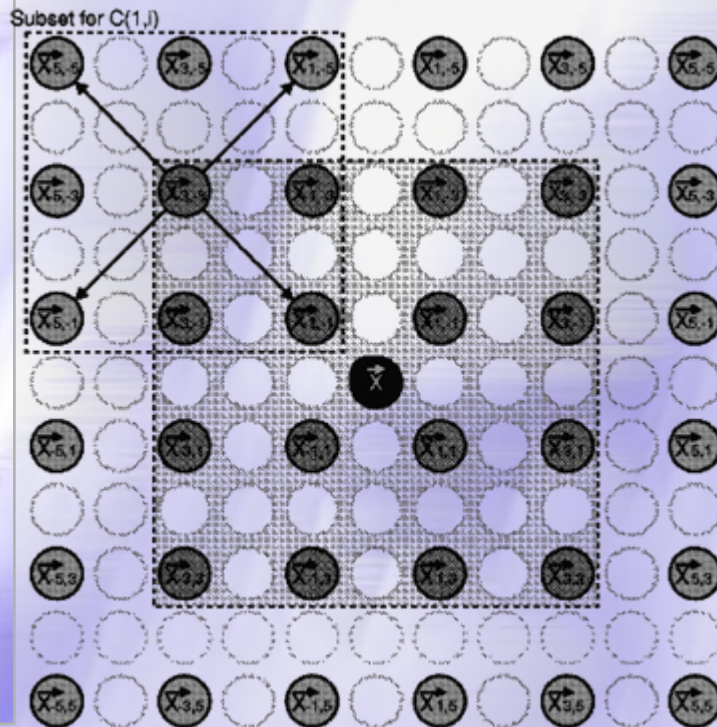
$$SSE = \left\| \vec{y} - \vec{\alpha} C \right\|_2$$

с матрицей C специального вида и находится его минимум



NEDI – локальное окно

$$\vec{y} = [\vec{x}_{-3,-3} \ \vec{x}_{-1,-3} \ \vec{x}_{1,-3} \ \vec{x}_{3,-3} \ \vec{x}_{-3,-1} \ \vec{x}_{-1,-1} \ \vec{x}_{1,-1} \ \vec{x}_{3,-1} \ \vec{x}_{-3,1} \ \vec{x}_{-1,1} \\ \vec{x}_{1,1} \ \vec{x}_{3,1} \ \vec{x}_{-3,3} \ \vec{x}_{-1,3} \ \vec{x}_{1,3} \ \vec{x}_{3,3}]$$



$$\left[\begin{array}{cccc} \vec{x}_{-5,-5} & \vec{x}_{-1,-5} & \vec{x}_{-5,-1} & \vec{x}_{-1,-1} \\ \vec{x}_{-3,-5} & \vec{x}_{1,-5} & \vec{x}_{-3,-1} & \vec{x}_{1,-1} \\ & & \dots & \\ \vec{x}_{1,1} & \vec{x}_{5,1} & \vec{x}_{1,5} & \vec{x}_{5,5} \end{array} \right]$$

NEDI - реализация

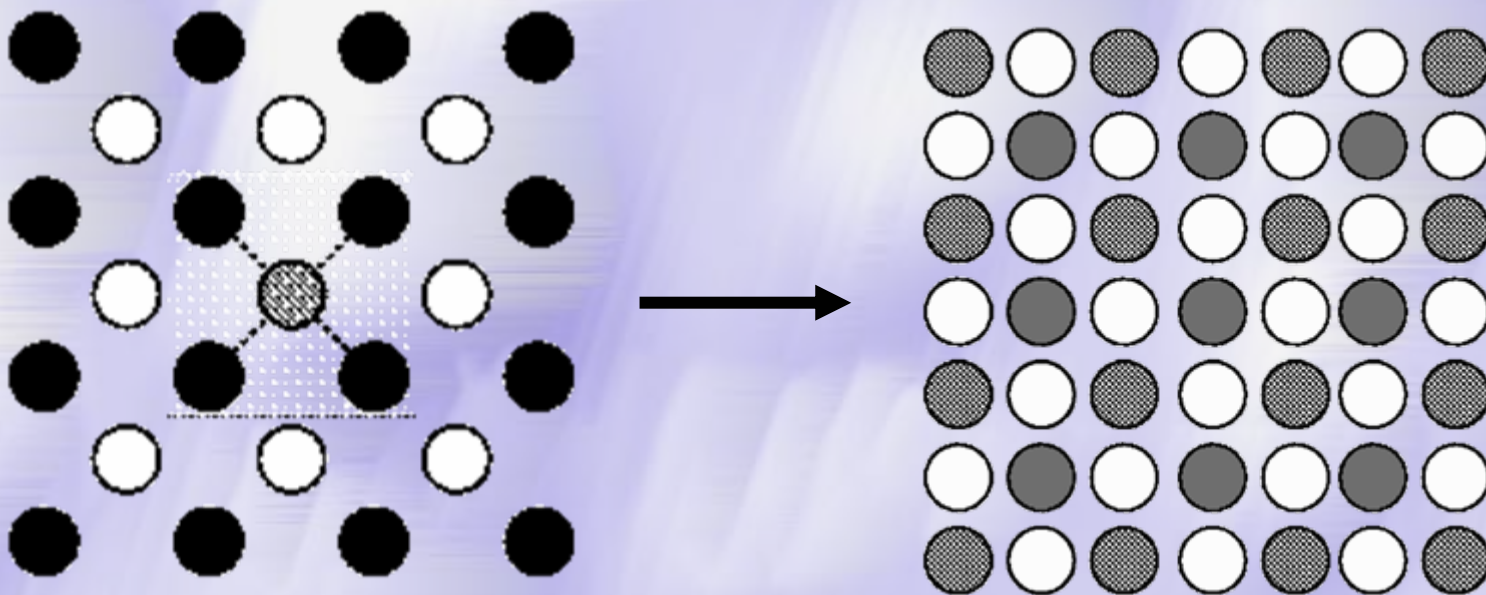
Минимум SSE находится как

$$\vec{\alpha} = (C^T C)^{-1} (C^T \vec{y})$$

Наиболее «тяжелая» операция во всем алгоритме – нахождение обратной матрицы, на что непосредственно влияет размер локального окна

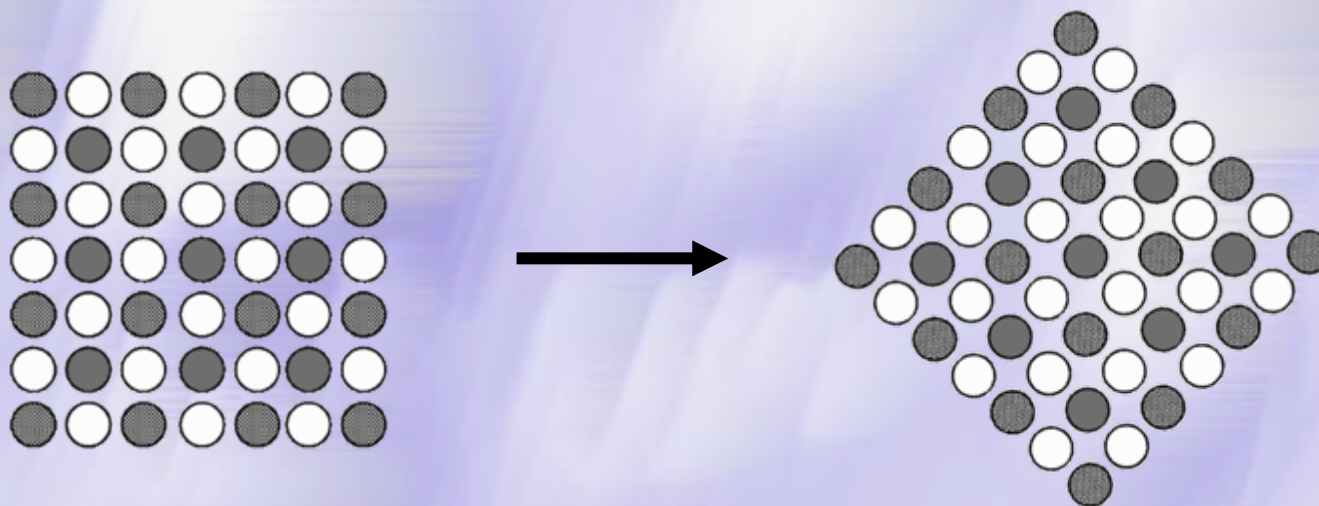
NEDI - реализация

Получили способ нахождения пикселей
со смещением $(2i+1, 2j+1)$



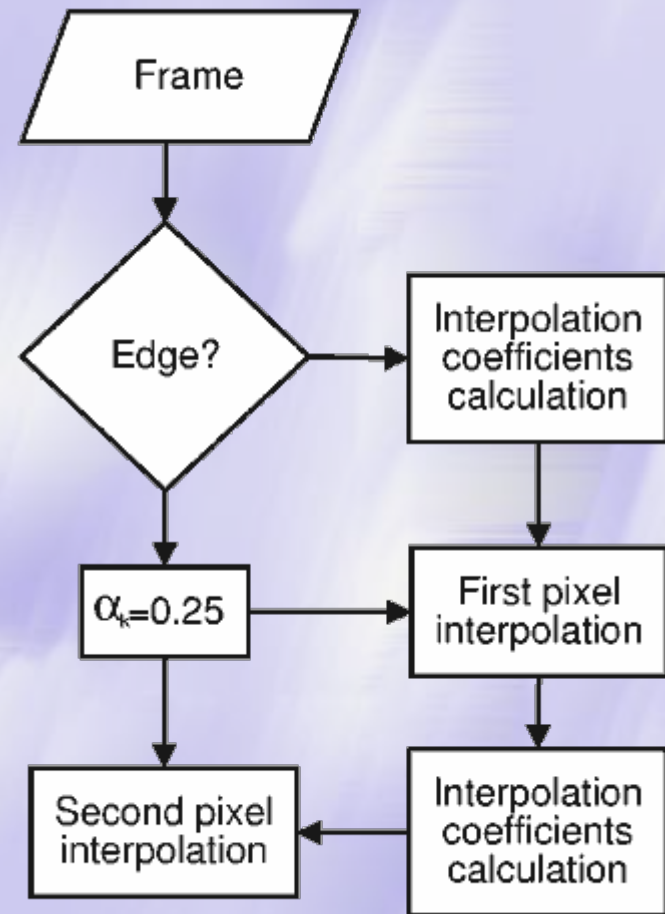
NEDI - реализация

Для нахождения остальных пикселей достаточно повернуть изображение на 45° алгоритм останется прежним

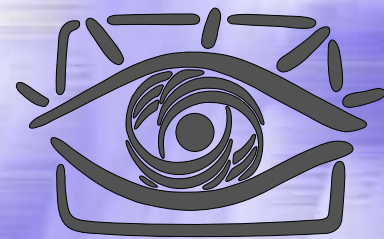


NEDI - СЛОЖНОСТЬ

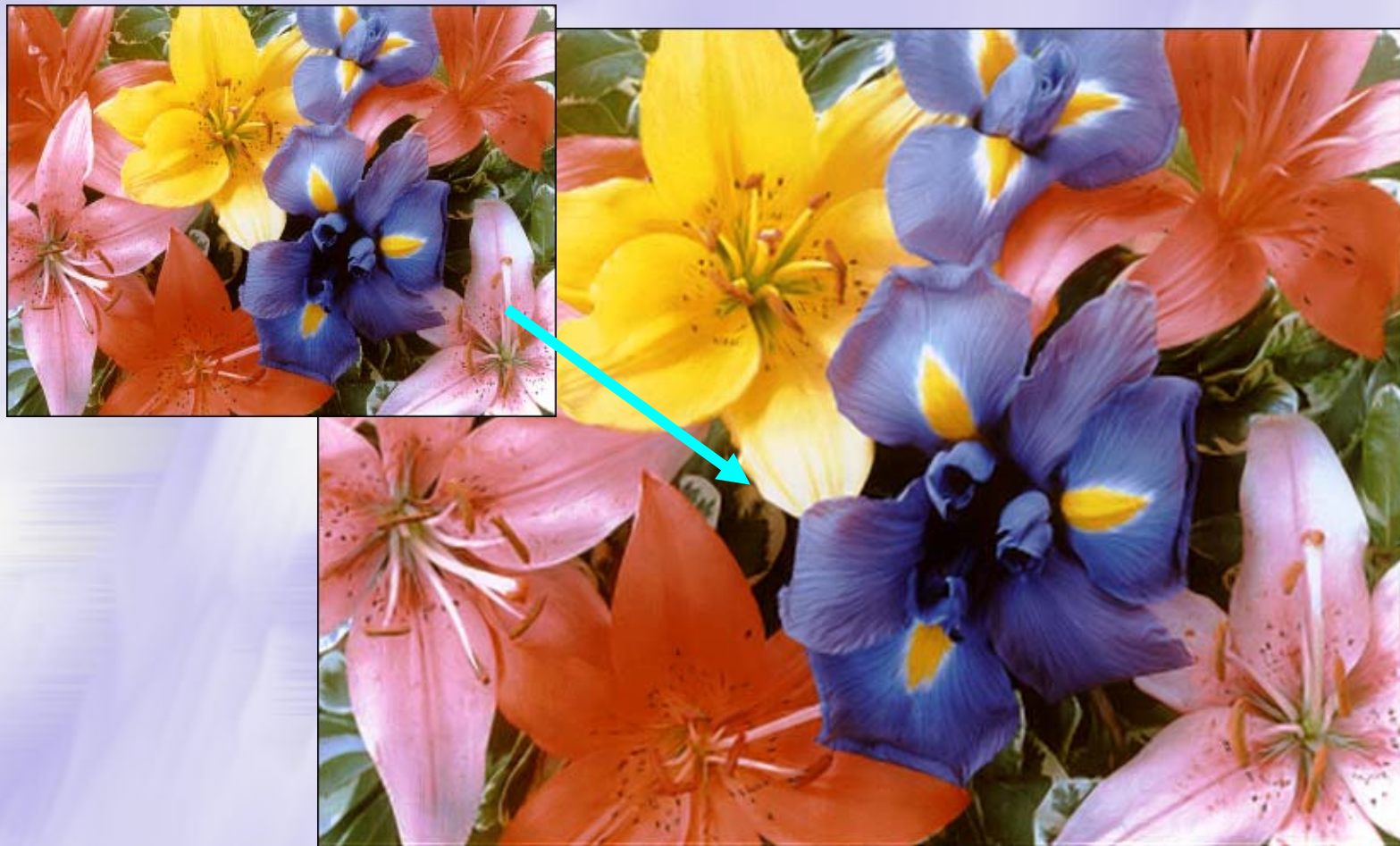
Уменьшение
количества операций
обращения матрицы



NEDI - результаты



GRAPHICS & MEDIA LAB
VIDEO GROUP



NEDI - результаты

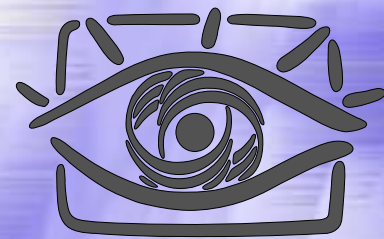
Bicubic



Nedi



NEDI - результаты



GRAPHICS & MEDIA LAB
VIDEO GROUP

Bicubic



Nedi



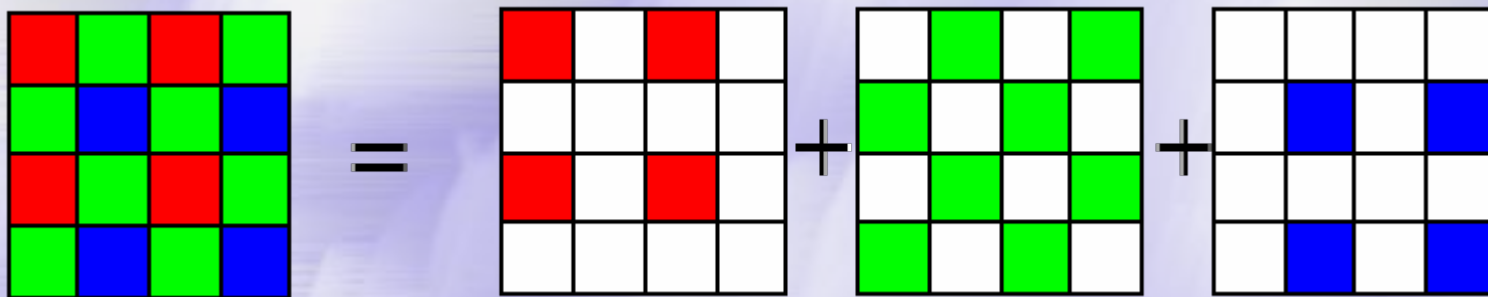
NEDI - проблемы

- ◆ Весьма медленный алгоритм требующий значительного ускорения
- ◆ Артефакты на краях изображения.



Применение NEDI

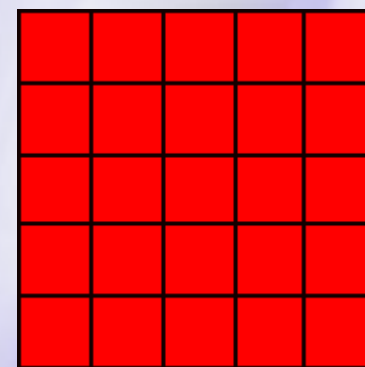
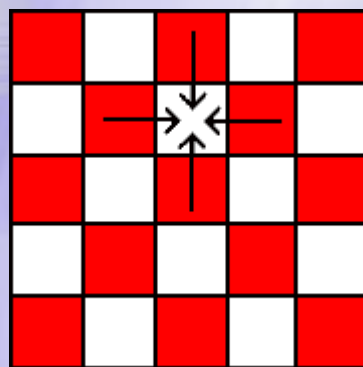
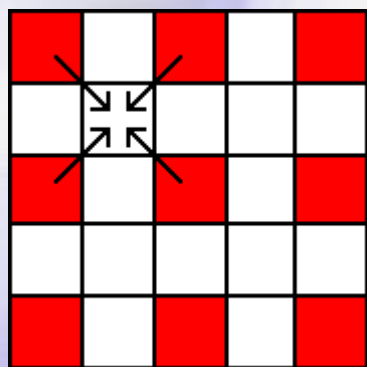
NEDI увеличивает изображение в 2 раза
что и требуется при восстановлении
изображения с CCD матрицы



Применение NEDI

Для синей и красной составляющей
требуются оба шага NEDI

1й шаг NEDI



Применение NEDI

Для зеленой составляющей требуются
только 2й шаг

