Цифрова обработка на изображения

Геометрични операции с изображения

доц. Милена Лазарова, кат. КС, ФКСУ

Билинейна интерполация

- Размери на оригиналното изображение *I*: $R' \times C'$
- Размери на мащабираното изображение *J*: $R \times C$
- Мащабиращи коефициенти (входно към изходно изобр.)

$$s_R = R' / R \text{ и } s_C = C' / C$$

- Ако $r_f = r \cdot s_R$ за $r = 1, \ldots, R$

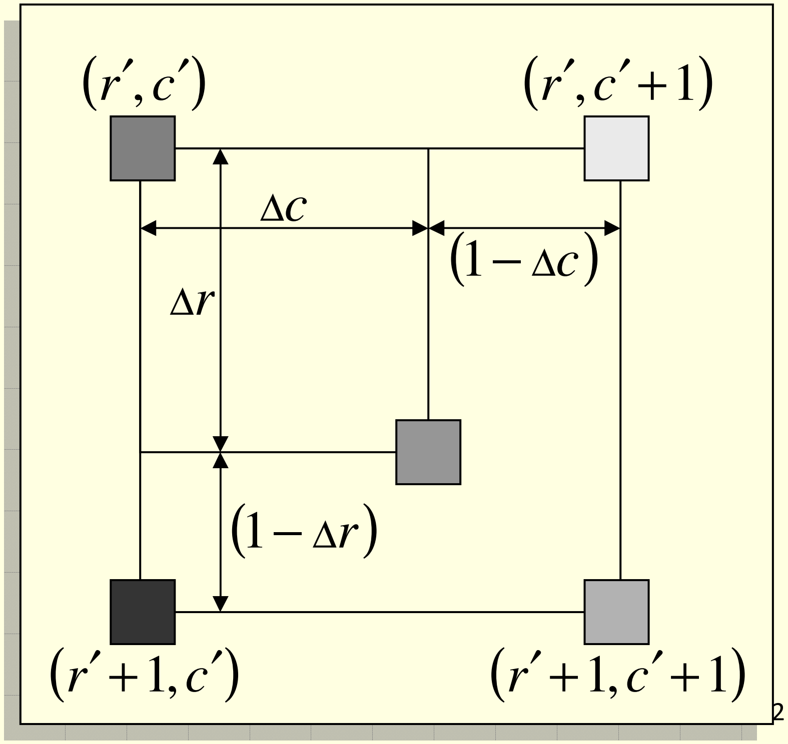
$$c_f = c \cdot s_C \text{ за } c = 1, \ldots, C$$

- и $r' = \lfloor r_f \rfloor$ и $c' = \lfloor c_f \rfloor$

$$\Delta r = r_f - r' \text{ и } \Delta c = c_f - c'$$

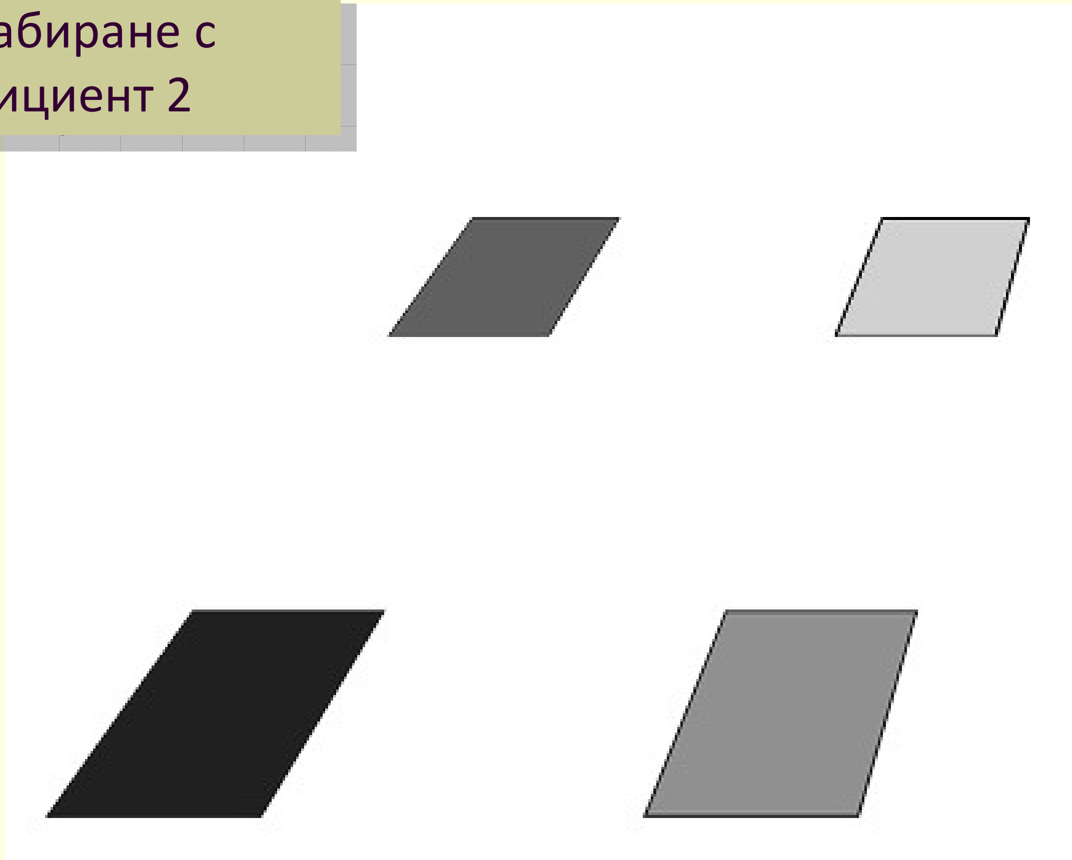
- то

$$\begin{aligned} J(r,c) = {} & I(r',c') \cdot (1-\Delta r) \cdot (1-\Delta c) \\ & + I(r'+1,c') \cdot \Delta r \cdot (1-\Delta c) \\ & + I(r',c'+1) \cdot (1-\Delta r) \cdot \Delta c \\ & + I(r'+1,c'+1) \cdot \Delta r \cdot \Delta c \end{aligned}$$



Билинейна интерполация

Мащабиране с
коефициент 2

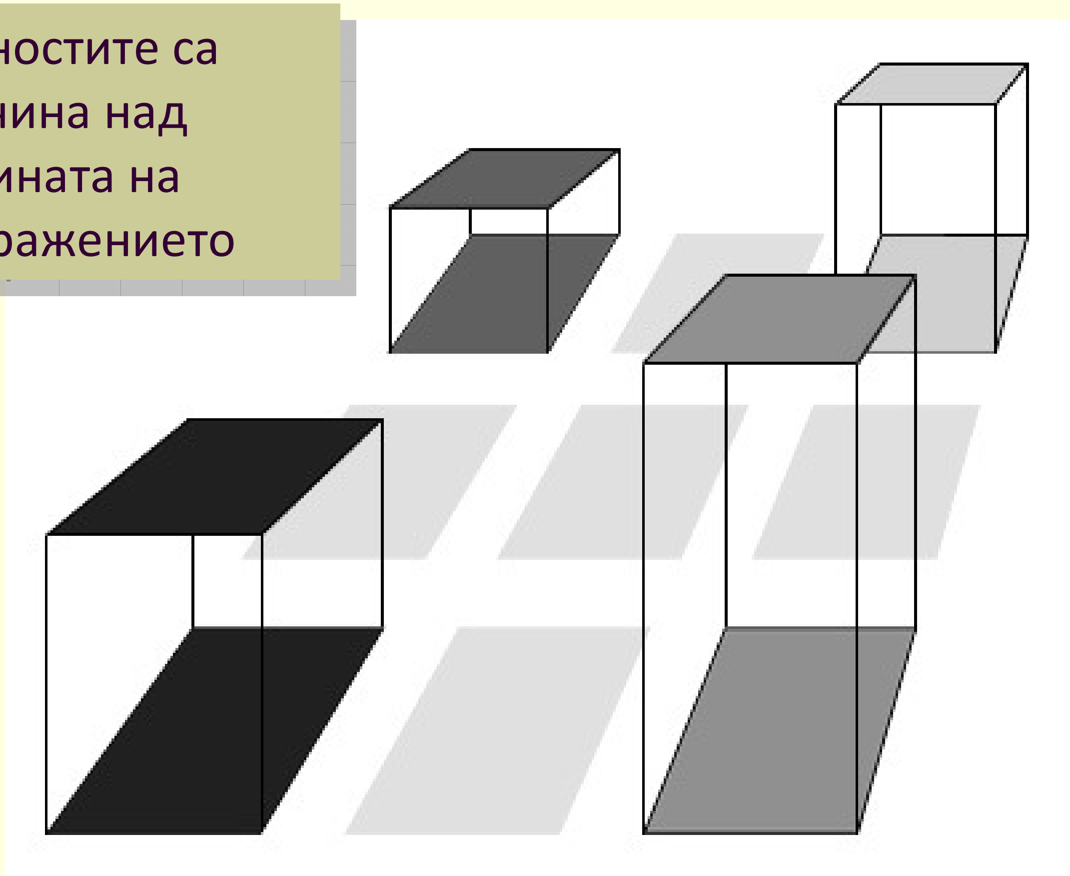


Билинейна интерполация

Мащабирането изисква да се добавят се нови пиксели

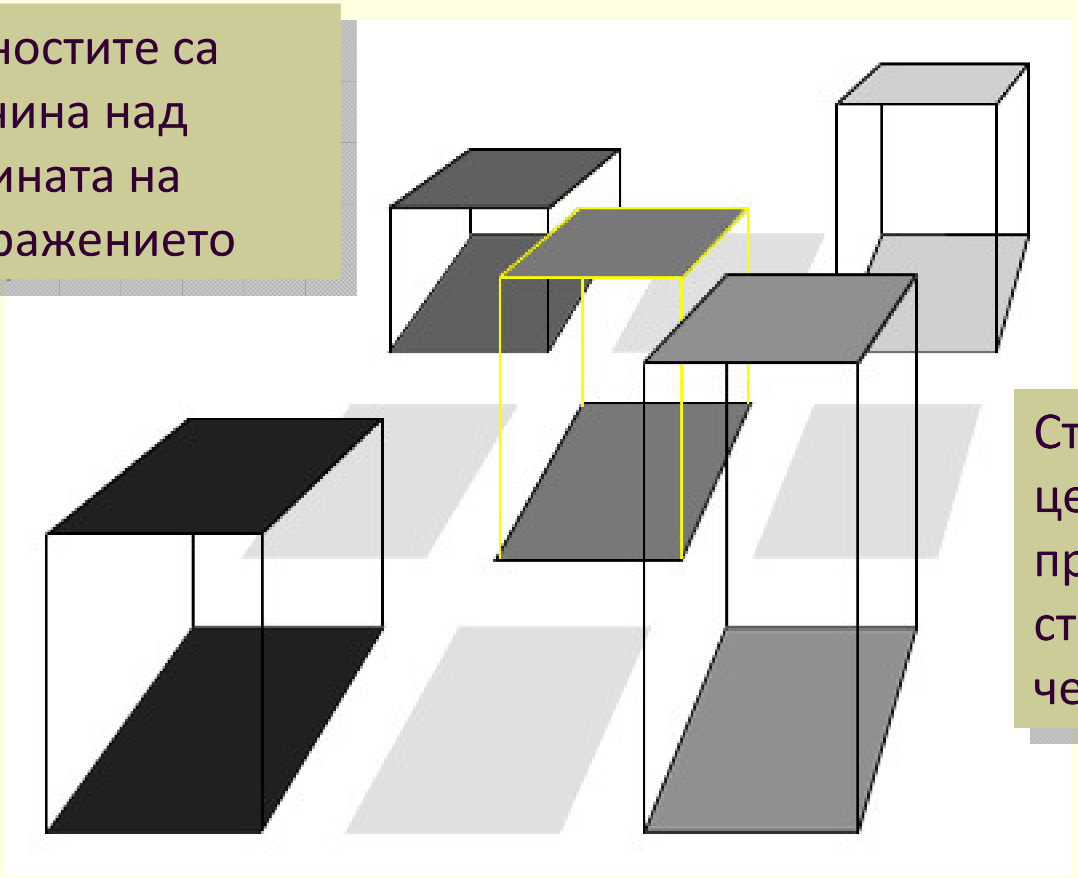
Билинейна интерполация

Стойностите са височина над равнината на изображението



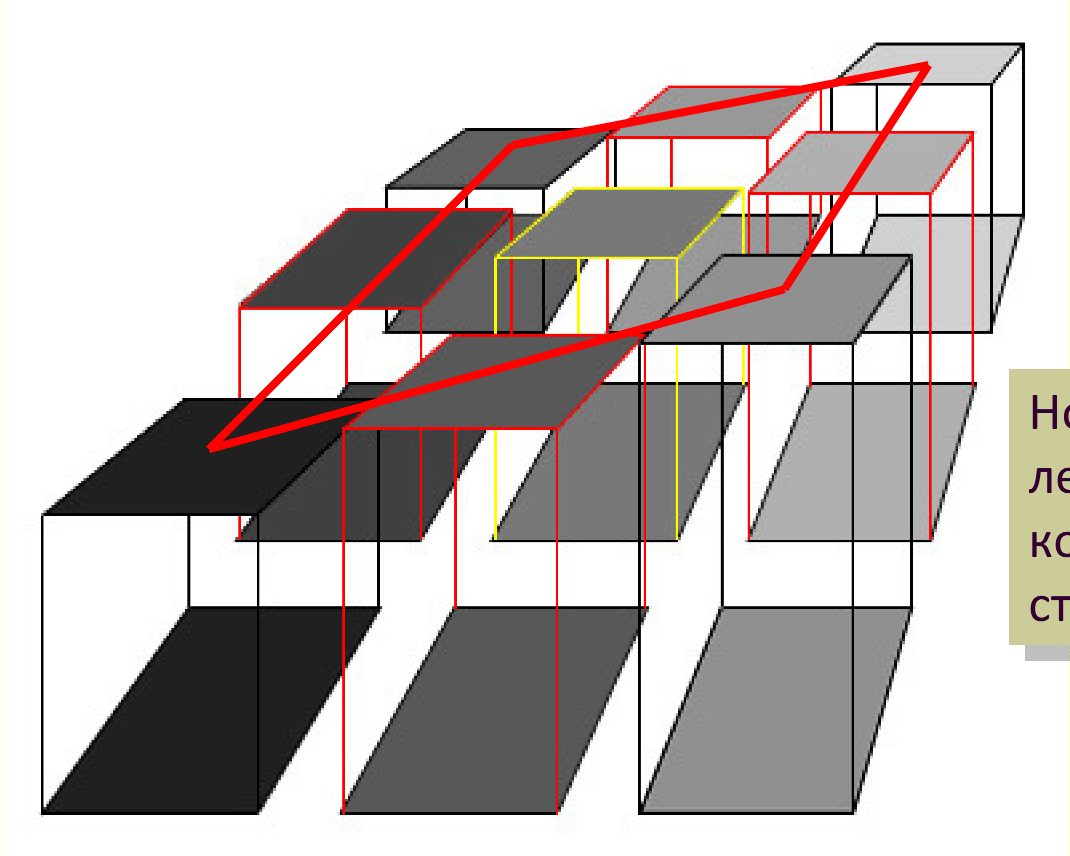
Билинейна интерполация

Стойностите са височина над равнината на изображението



Стойността в центъра е претеглена средна стойност от четирите ъгъла

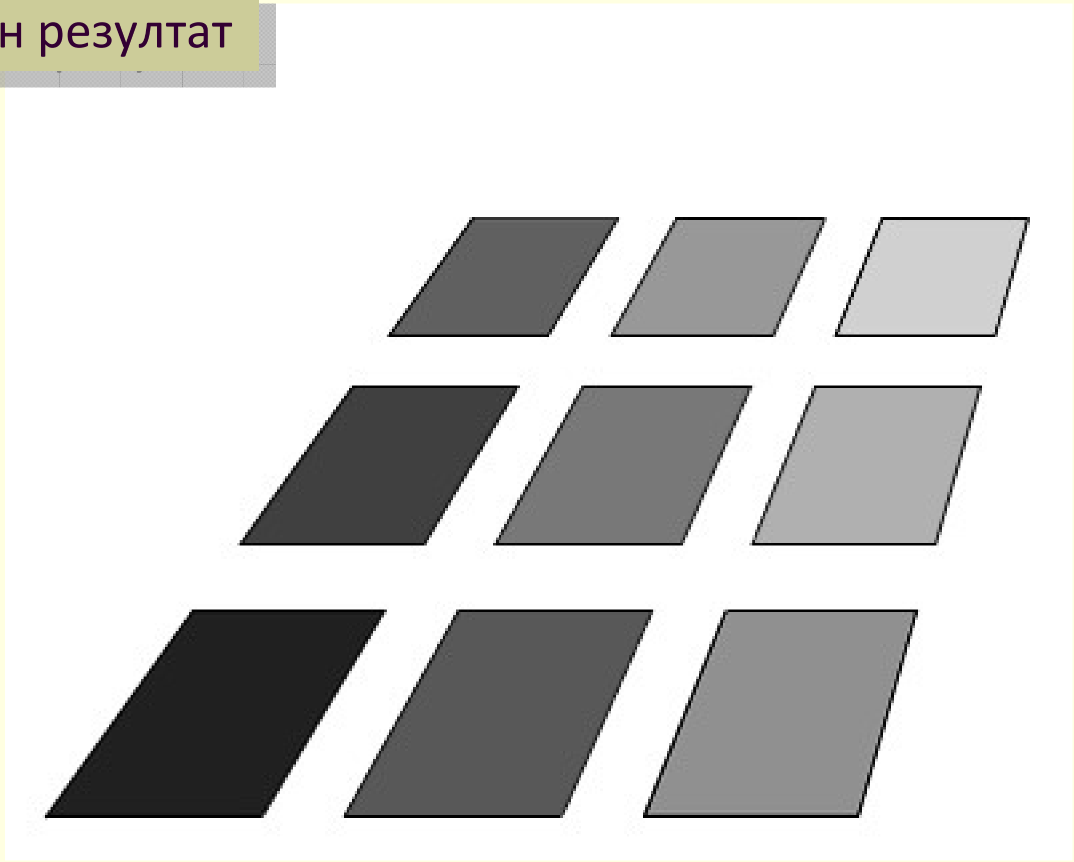
Билинейна интерполация



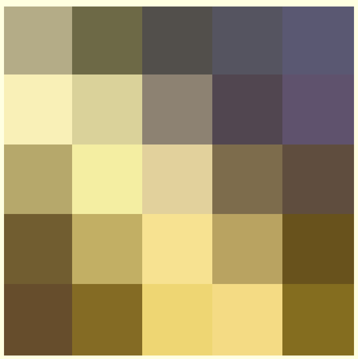
Новите пиксели лежат на линиите, които свързват старите стойности

Билинейна интерполация

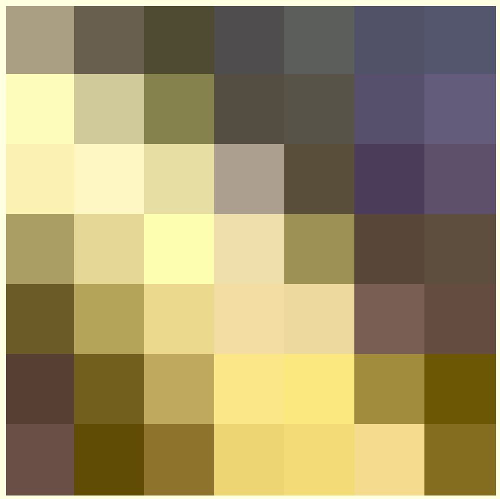
Краен резултат



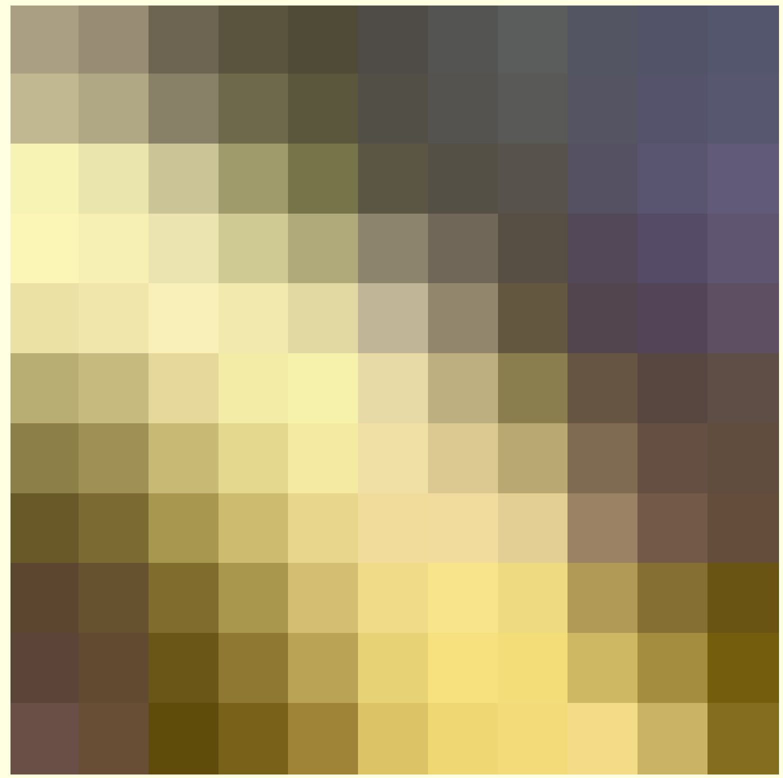
Билинейна интерполация



5:7



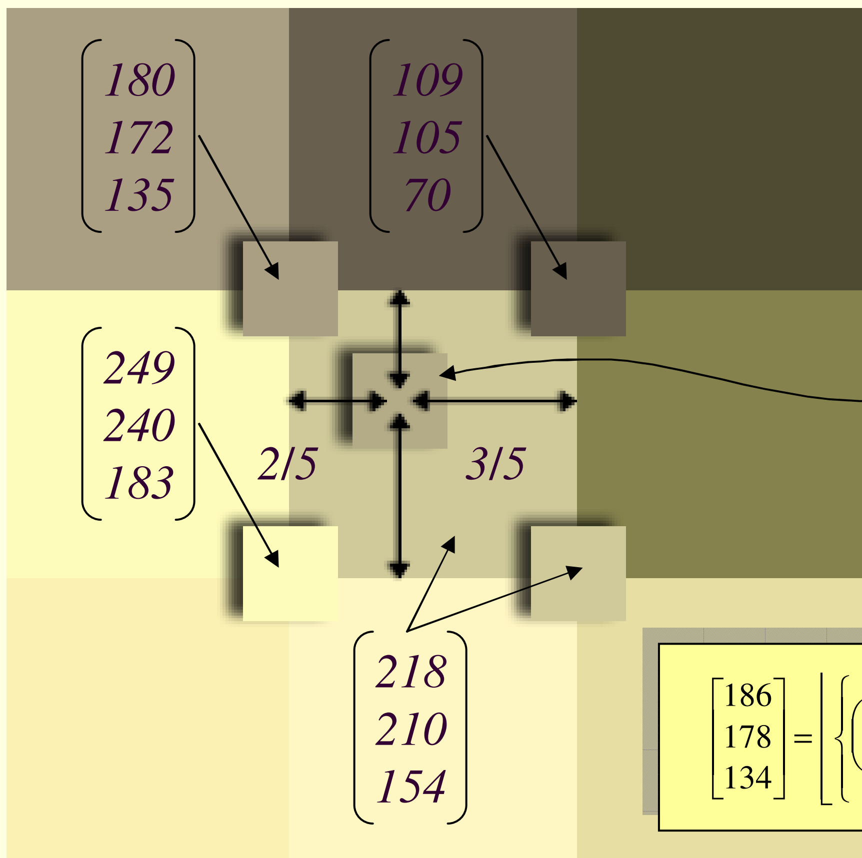
1:1

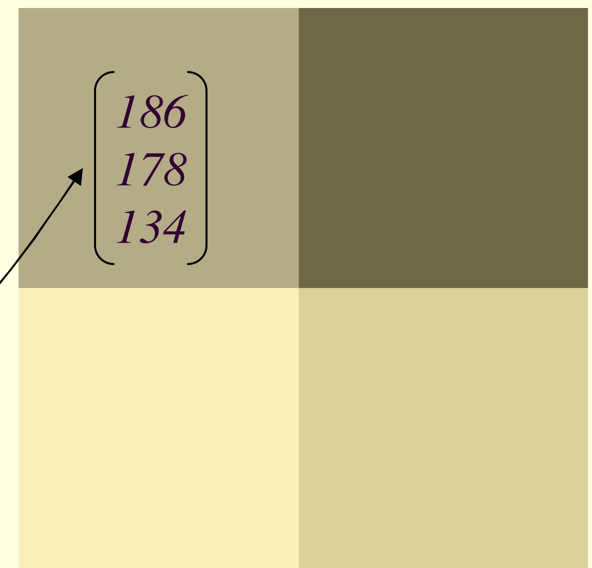


11:7

Билинейна интерполация

Пример: мащабиране на изображение до 5/7 от размерите му





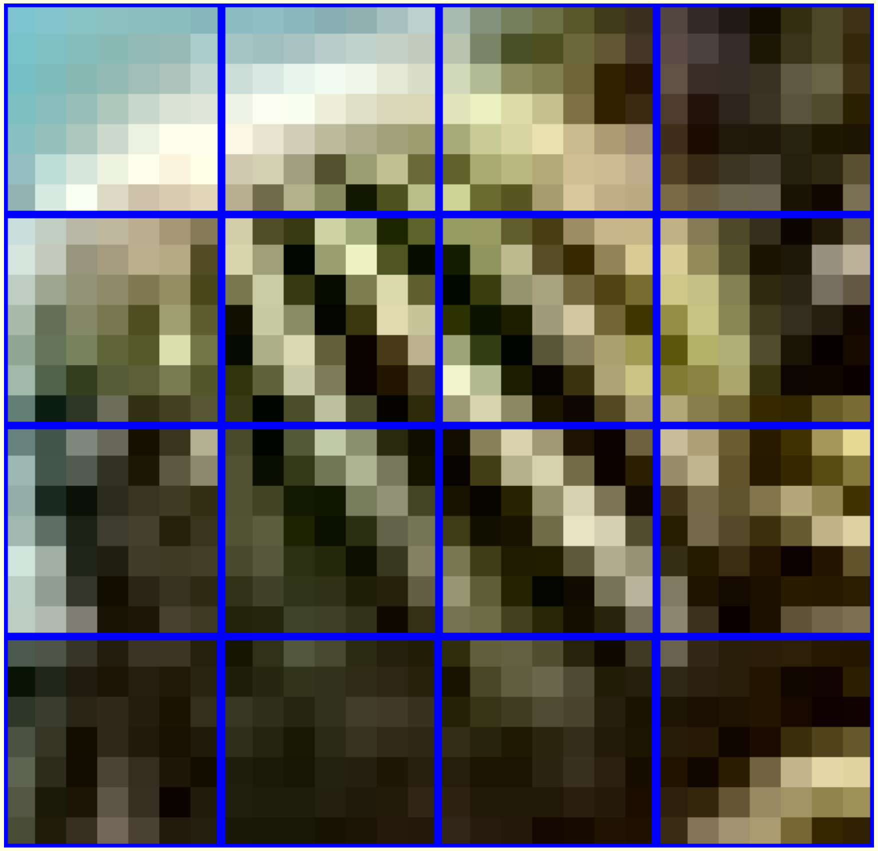
$$\begin{bmatrix}186\\178\\134\end{bmatrix} = \left\lfloor \left\{ \left(\frac{3}{5}\right)^2 \begin{bmatrix}180\\172\\135\end{bmatrix} + \left(\frac{2}{5}\right)\left(\frac{3}{5}\right) \begin{bmatrix}249\\240\\183\end{bmatrix} + \left(\frac{3}{5}\right)\left(\frac{2}{5}\right) \begin{bmatrix}109\\105\\70\end{bmatrix} + \left(\frac{2}{5}\right)^2 \begin{bmatrix}218\\210\\154\end{bmatrix} \right\} + 0.5 \right\rfloor$$

Билинейна интерполация

Пример: мащабиране на изображение до 3/7 от размерите на оригиналното изображение

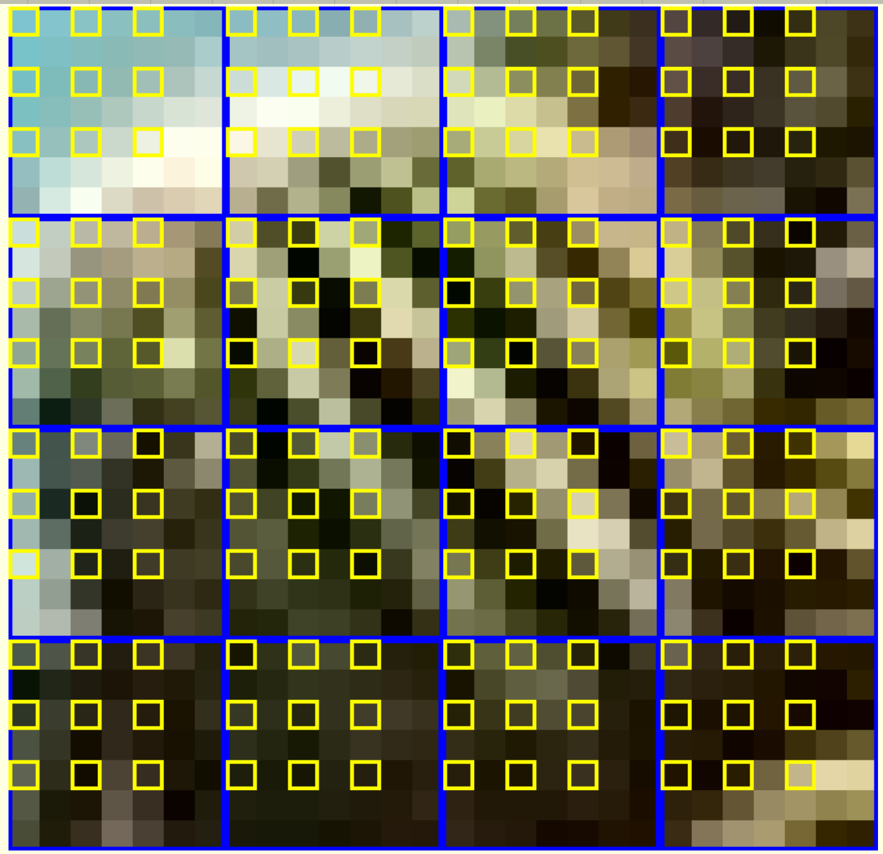
Билинейна интерполация

За всяка област с размери 7×7 се избират 3×3 = 9 координати на пиксели



Билинейна интерполация

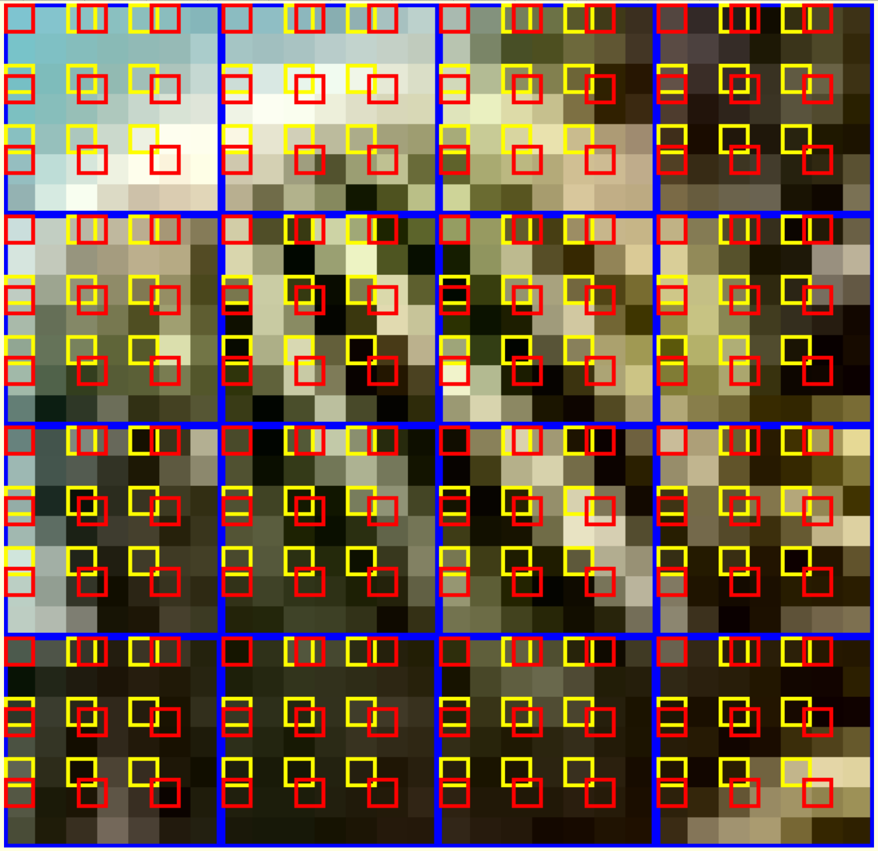
При мащабиране с най-близки съседи 9 пиксела са използваните стойности от оригиналното изображение



В жълто: пикселите, използвани при метода “най-близки съседи”

Билинейна интерполация

При мащабиране с билинейна интерполация 9те използвани стойности са равномерно разпределени



В жълто: пикселите, използвани при метода “най-близки съседи”

В червено: пиксели, използвани при метода “билинейна интерполация”

Билинейна интерполация

При мащабиране с билинейна интерполация 9те използвани стойности са равномерно разпределени

В жълто: пикселите, използвани при метода “най-близки съседи”

В червено: пиксели, използвани при метода “билинейна интерполация”

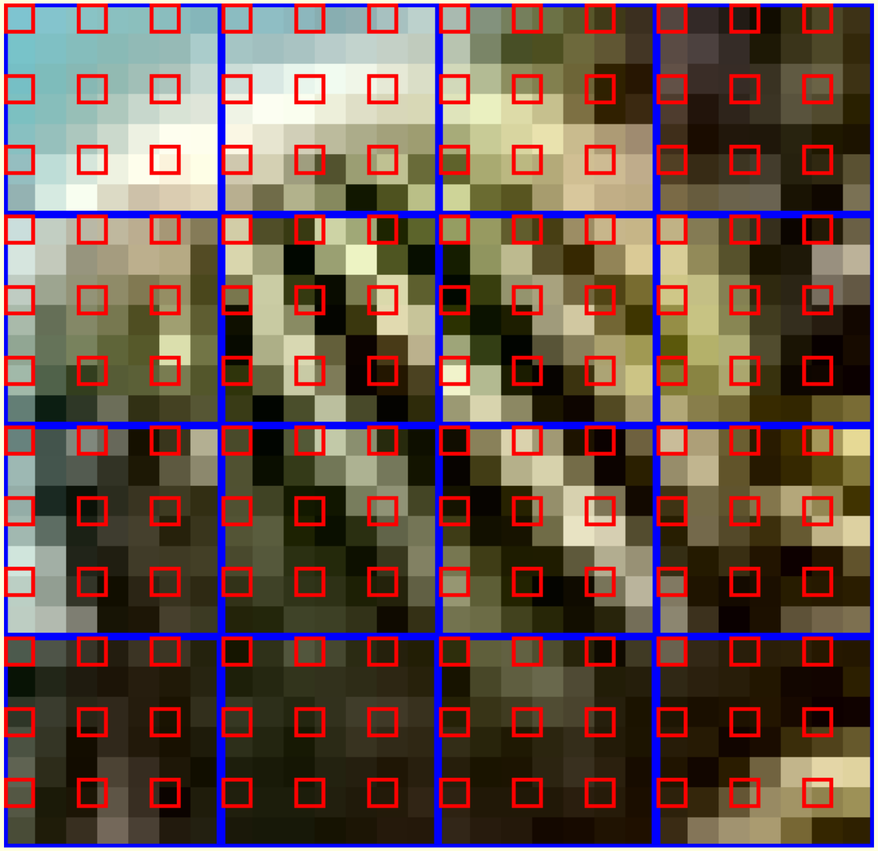
Билинейна интерполация

При мащабиране с билинейна интерполация 9те използвани стойности са равномерно разпределени

Избраните стойности покриват пиксели в изображението

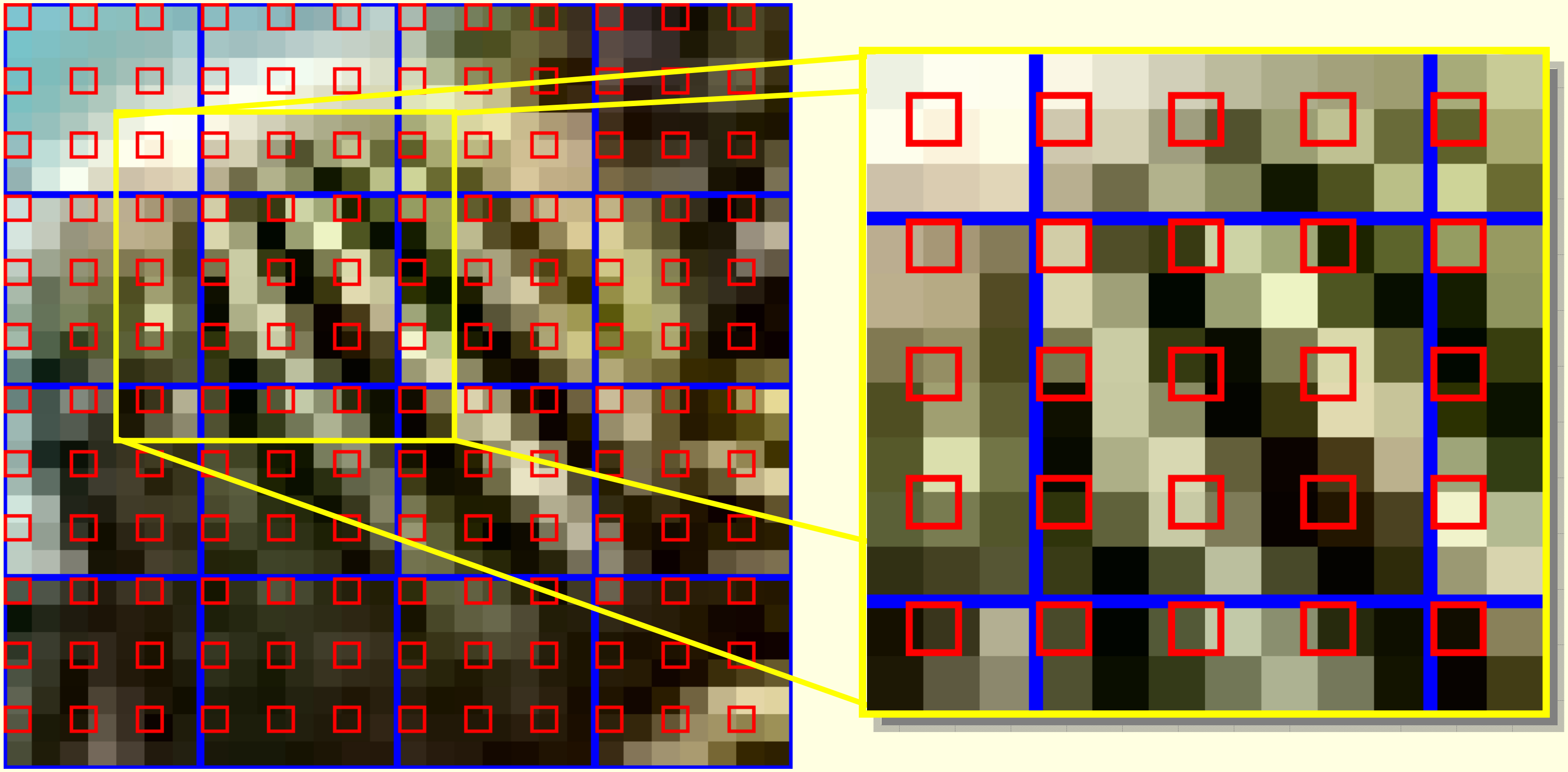
Билинейна интерполация

За всяка област с размери 7×7 се избират 3×3 = 9 координати на пиксели



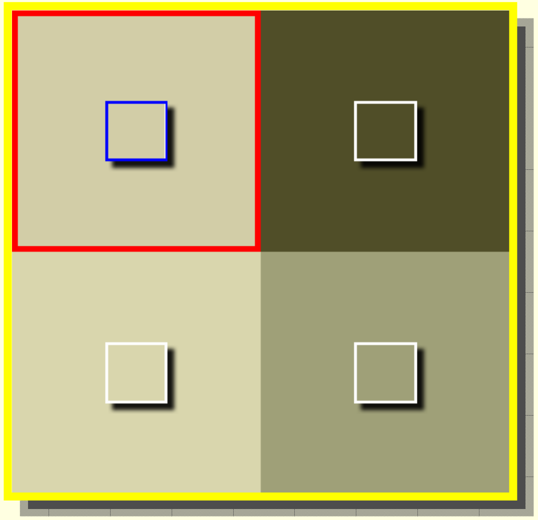
Избраните стойности покриват пиксели в изображението

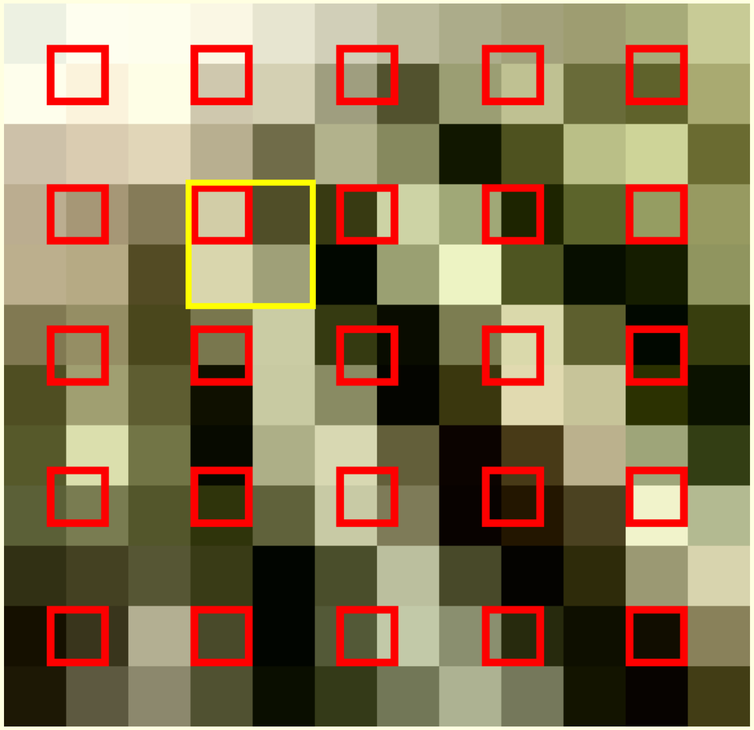
Билинейна интерполация



Билинейна интерполация

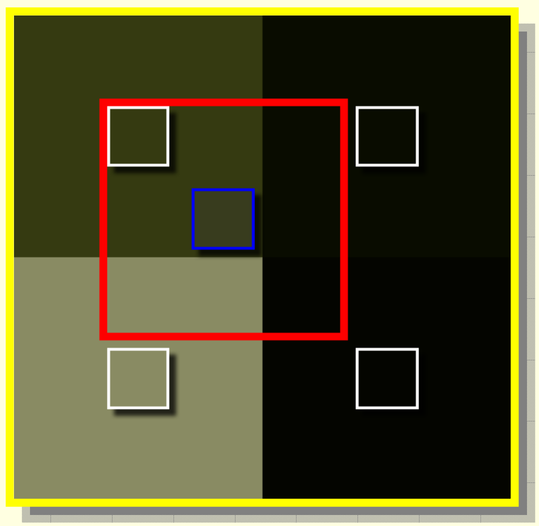
В син цвят: местоположение на пиксел в изходното изображение

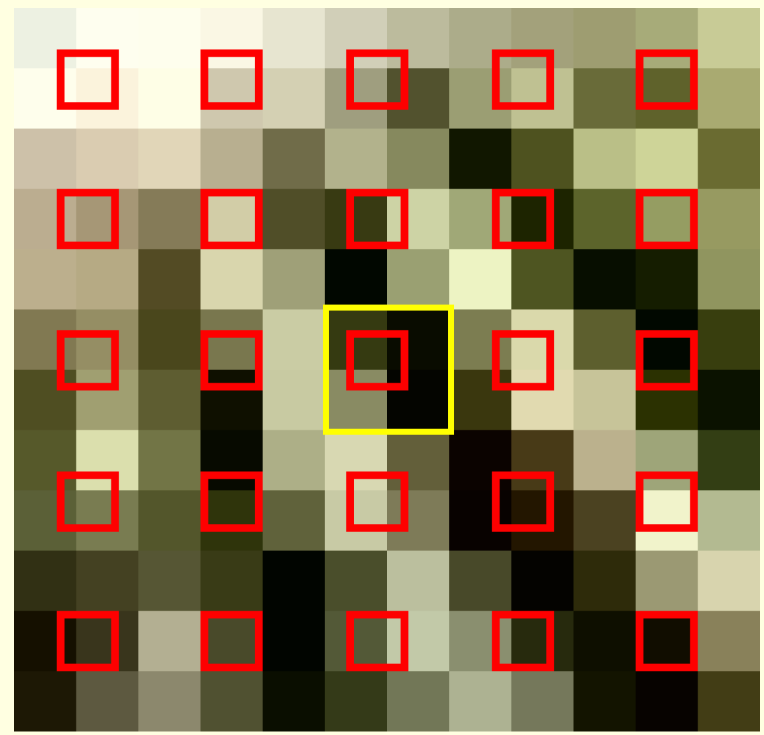




Билинейна интерполация

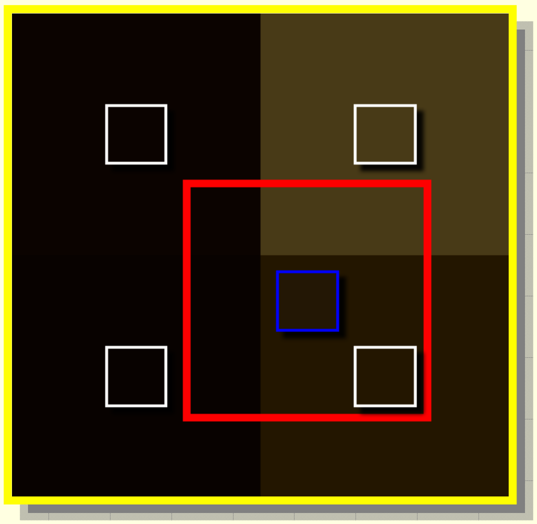
В син цвят: местоположение на пиксел в изходното изображение

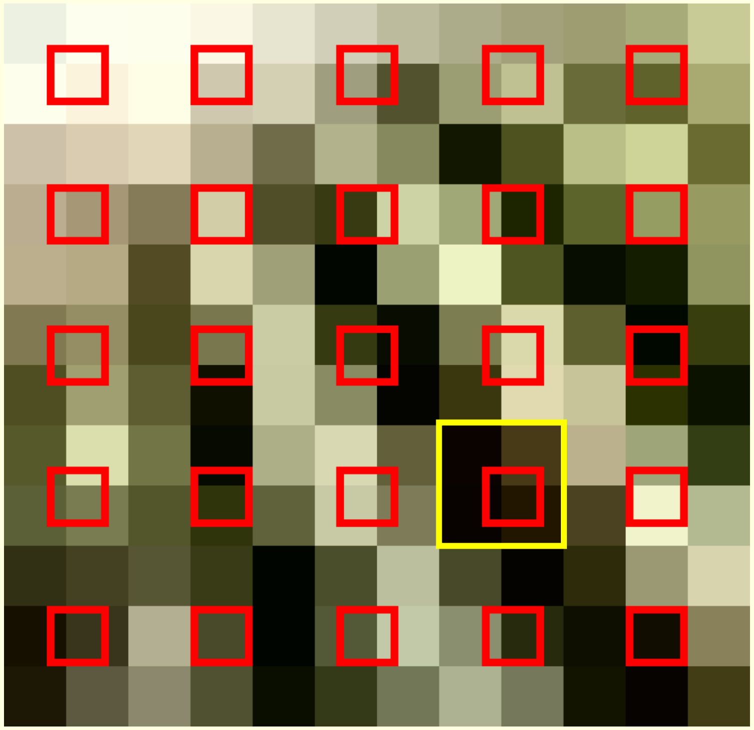




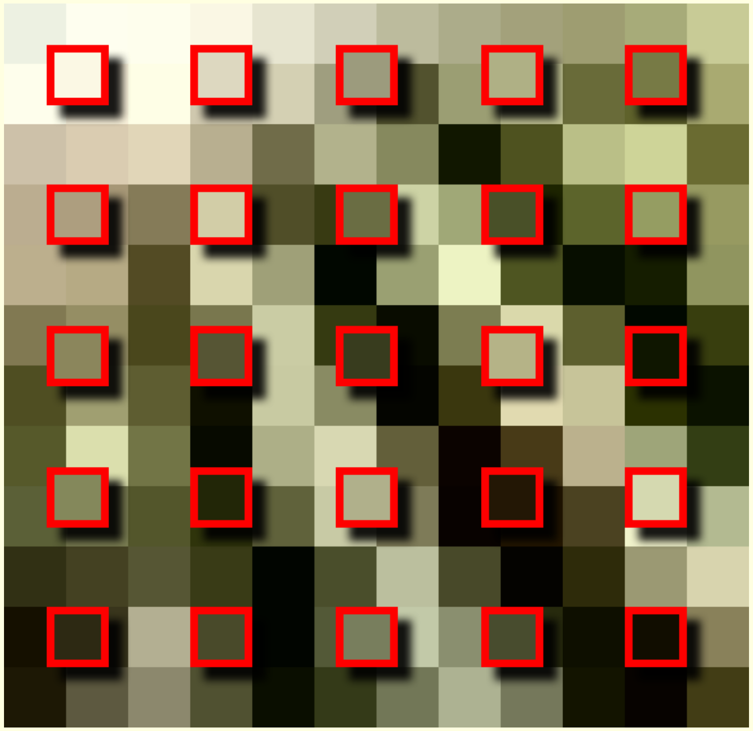
Билинейна интерполация

В син цвят: местоположение на пиксел в изходното изображение

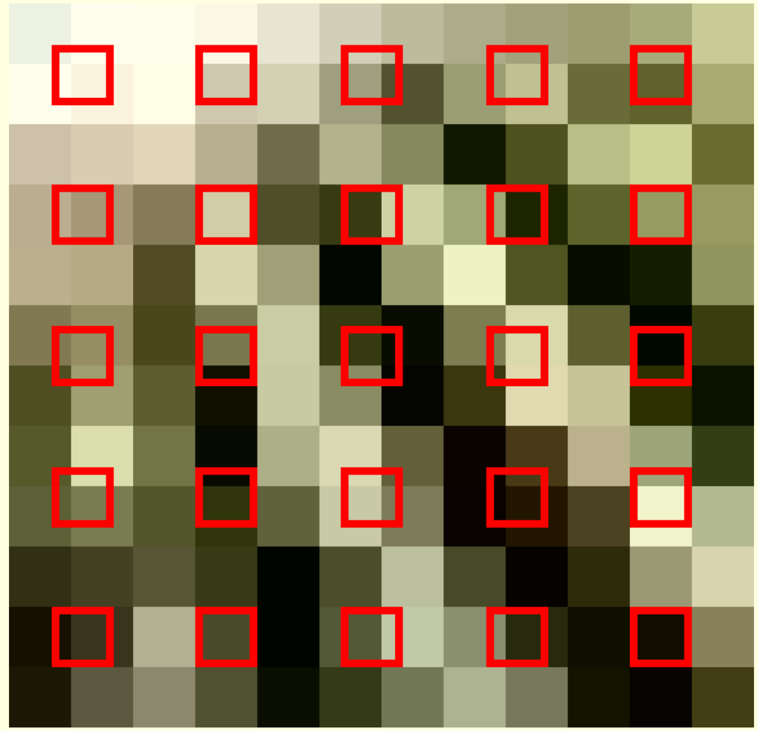




Билинейна интерполация

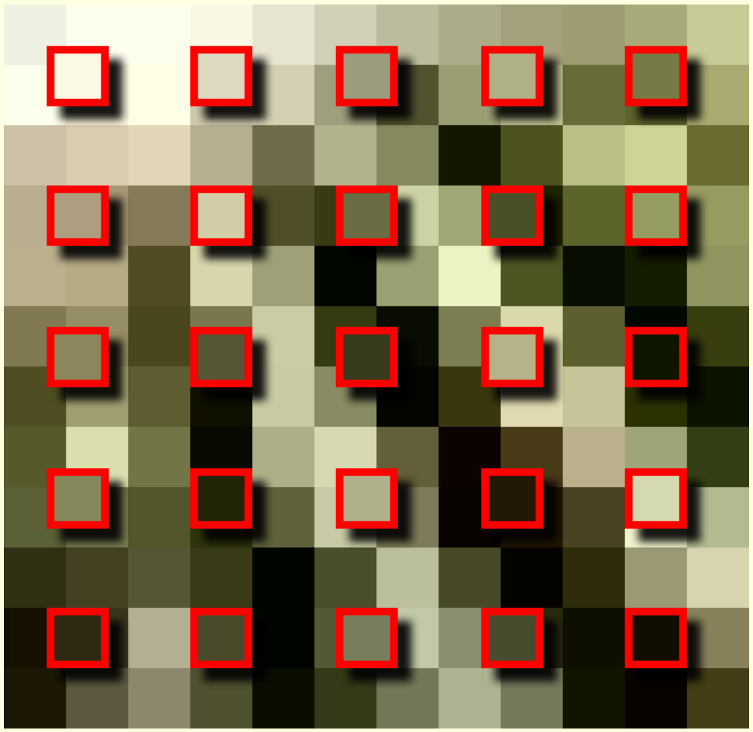


местоположение и стойност
на новите пиксели



местоположение на
новите пиксели

Билинейна интерполация



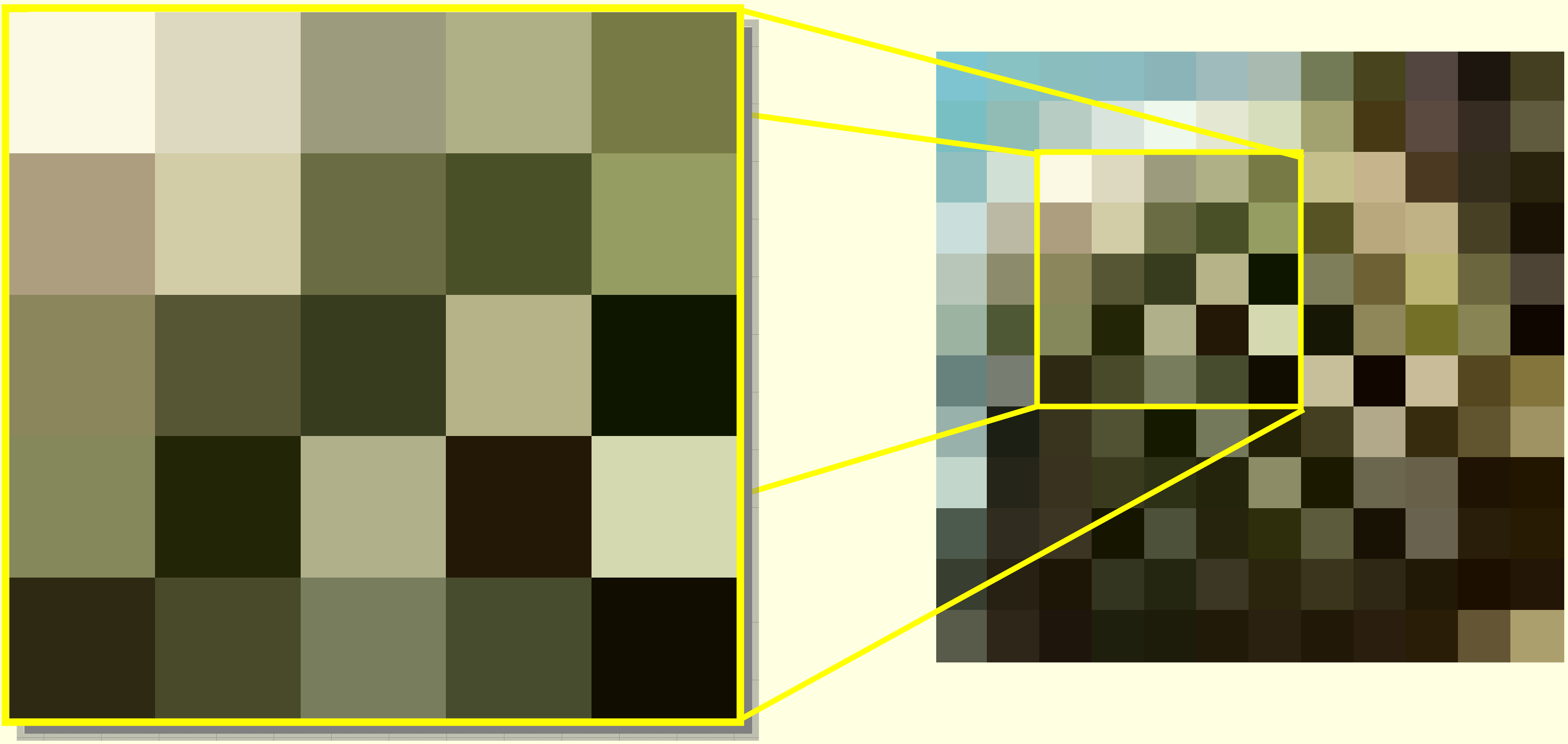
местоположение и стойност
на новите пиксели

местоположение и стойност
на новите пиксели

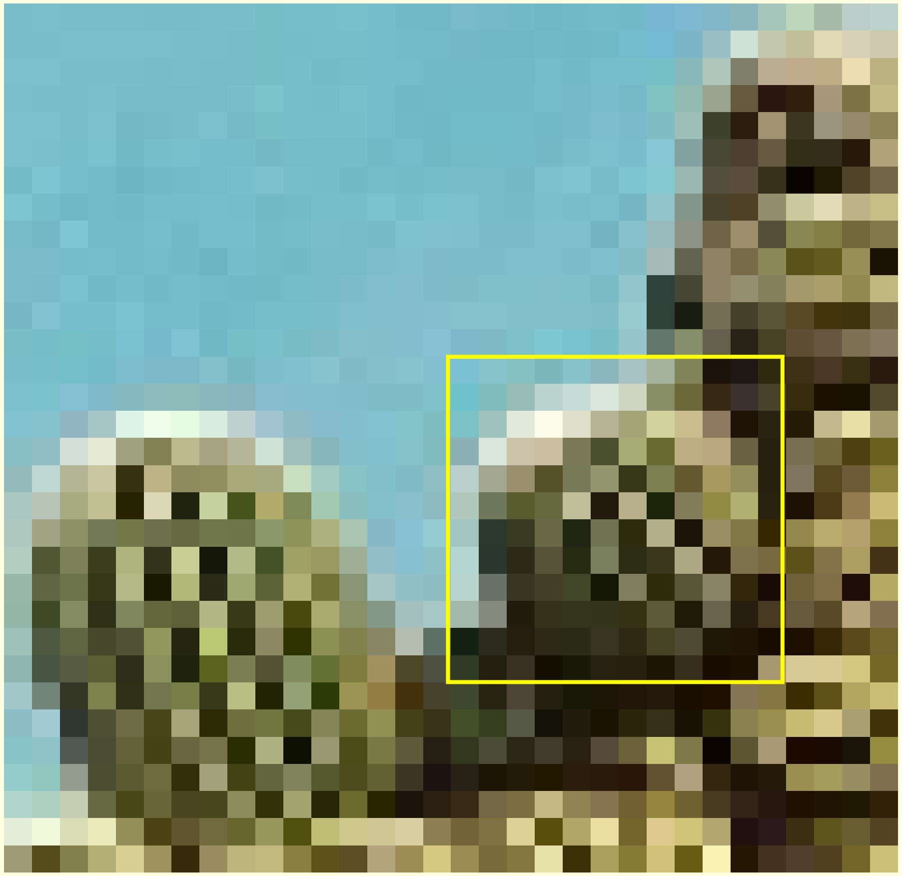
Билинейна интерполация

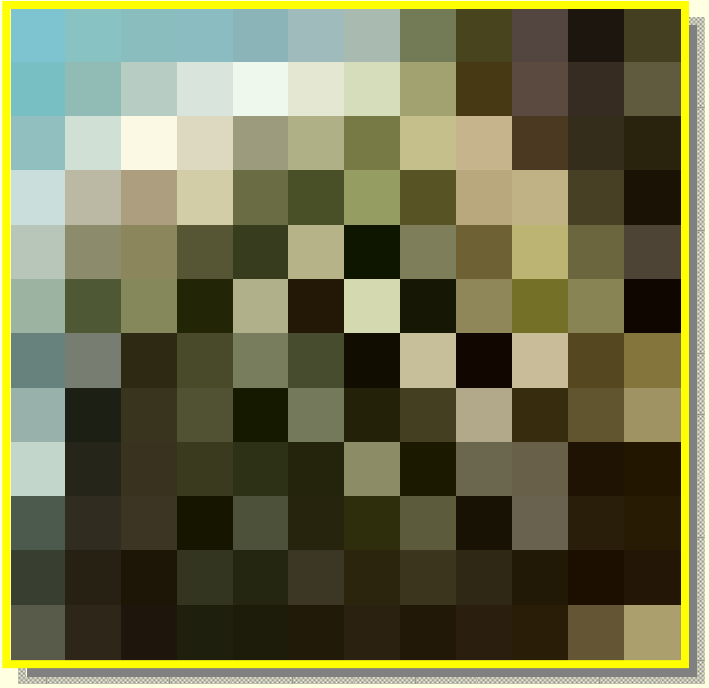
ново изображение от
новите пиксели

Билинейна интерполация

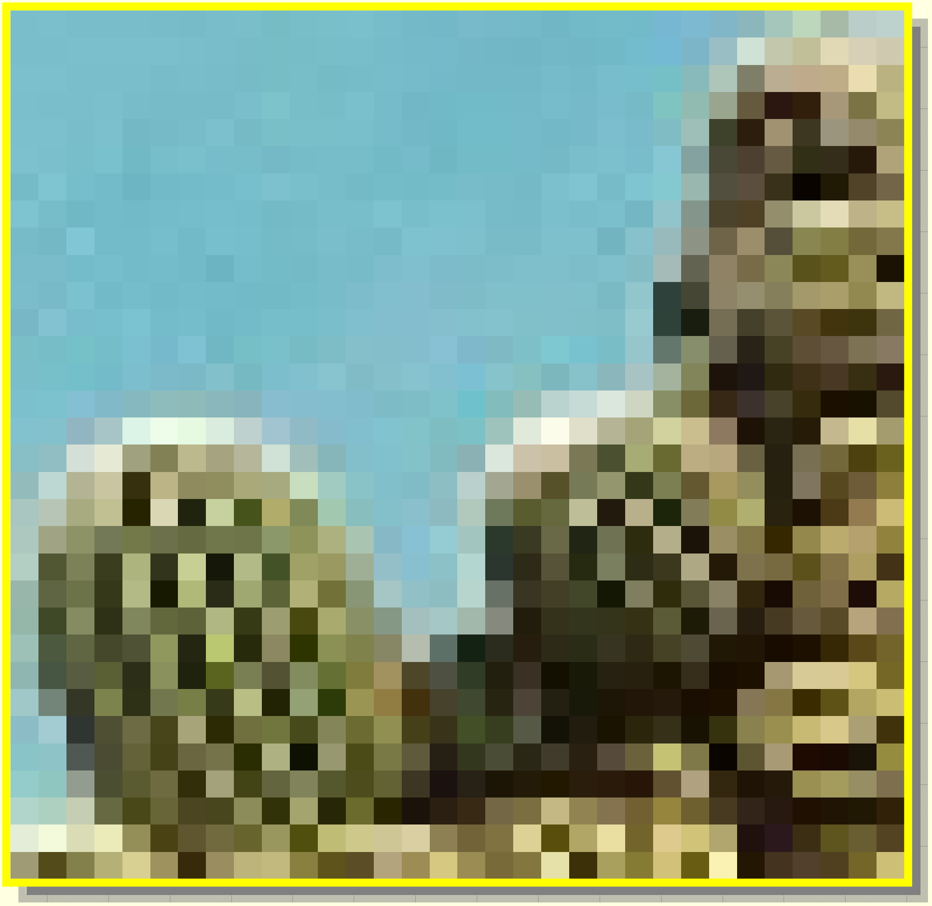


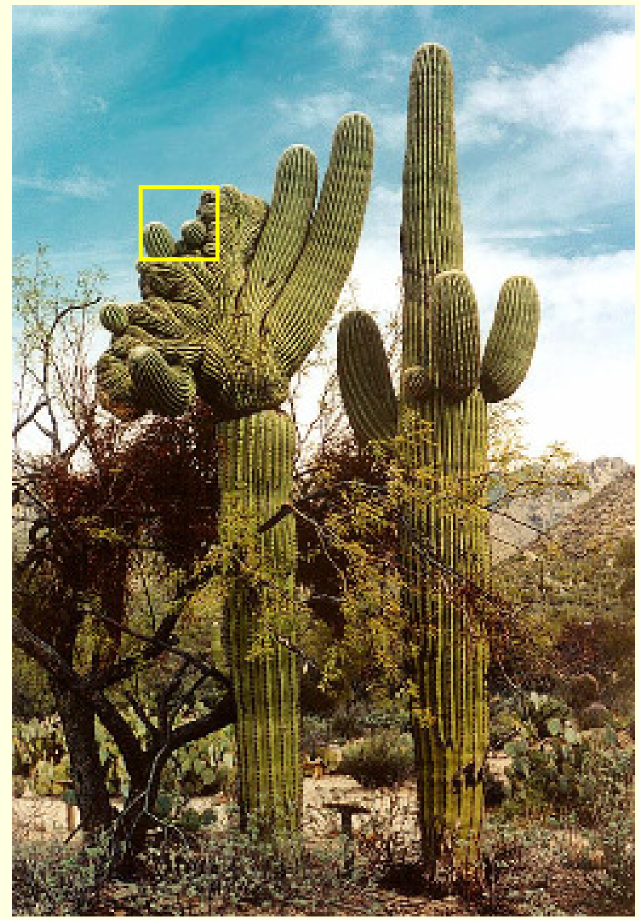
Билинейна интерполация





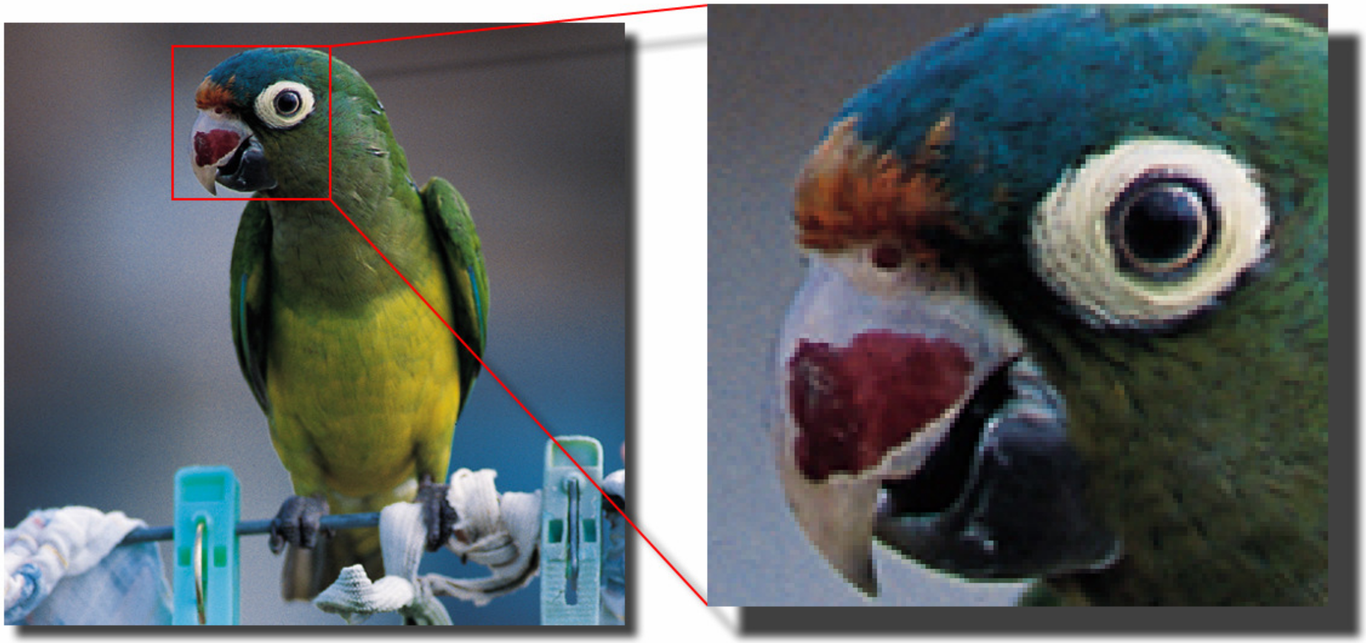
Билинейна интерполация





Билинейна интерполация - пример

Мащабиране:
увеличаване 4×

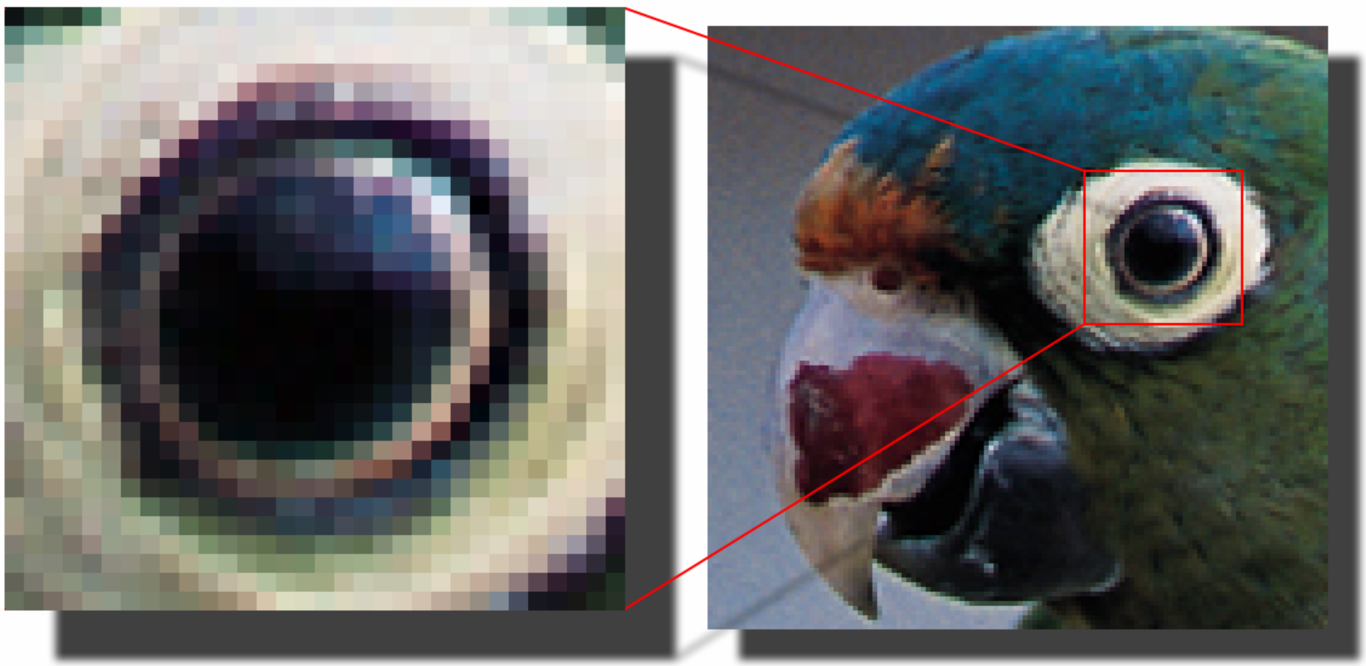


оригинално изображение

част от изображението

Билинейна интерполация - пример

Мащабиране:
увеличаване 4×



пример за интерполация

част от изображението

Билинейна интерполация - пример

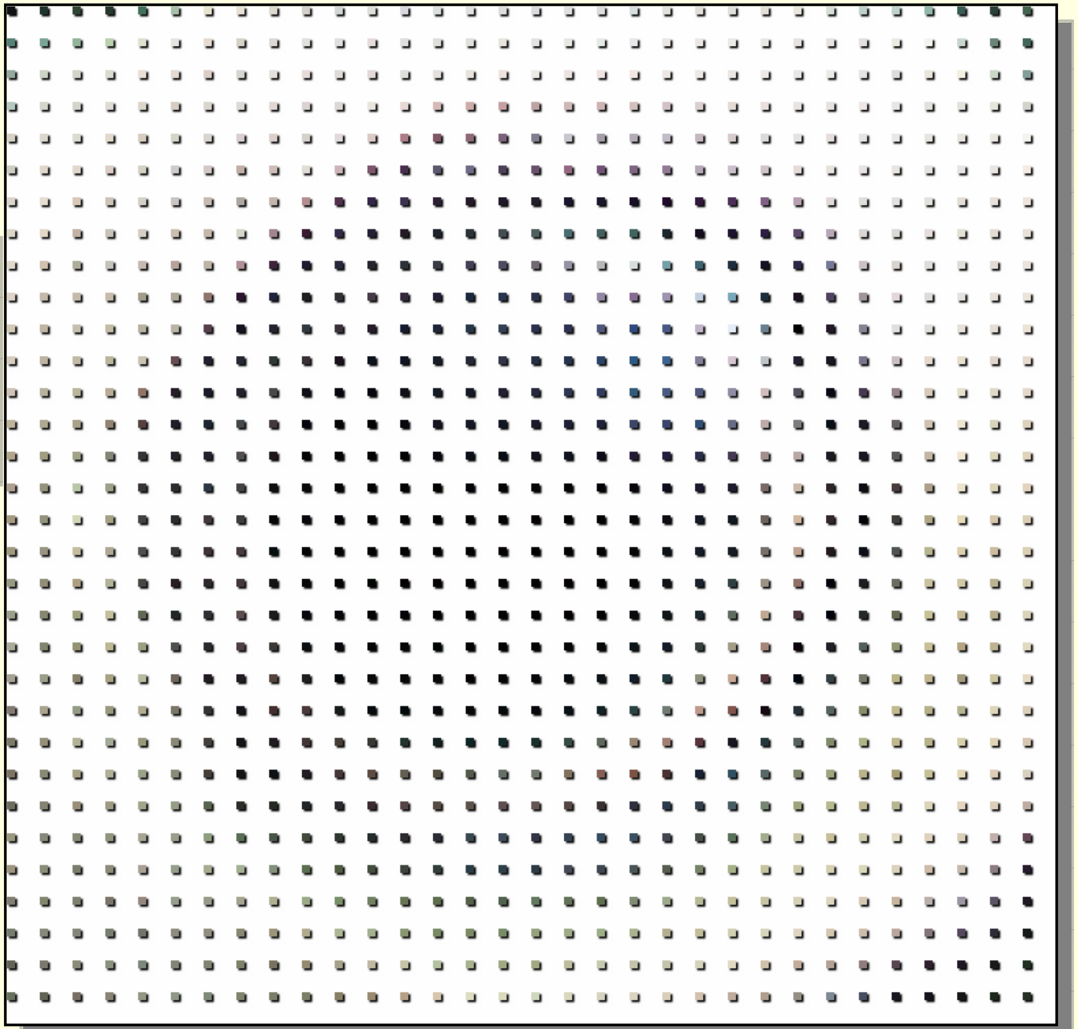
Мащабиране: увеличаване 4×

Създава се изображение с размери, четири пъти по-големи от тези на оригиналното изображение

Билинейна интерполация - пример

Мащабиране:
увеличаване 4×

Всеки 4-ти пиксел
на всеки 4-ти ред има
стойността на пиксел от
оригиналното изображение

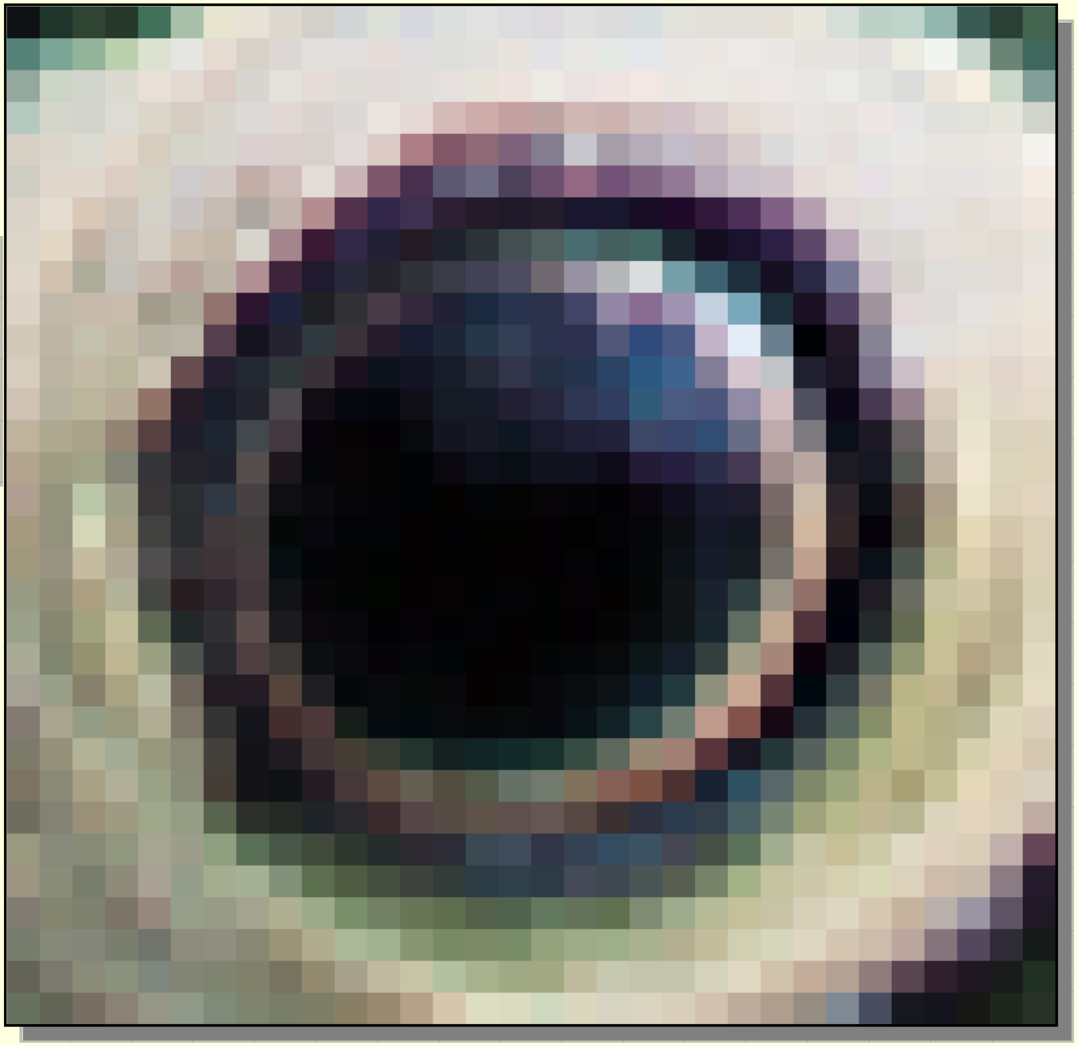


Билинейна интерполация - пример

Мащабиране: увеличаване 4×

Всеки 4-ти пиксел
на всеки 4-ти ред има
стойността на пиксел от
оригиналното изображение

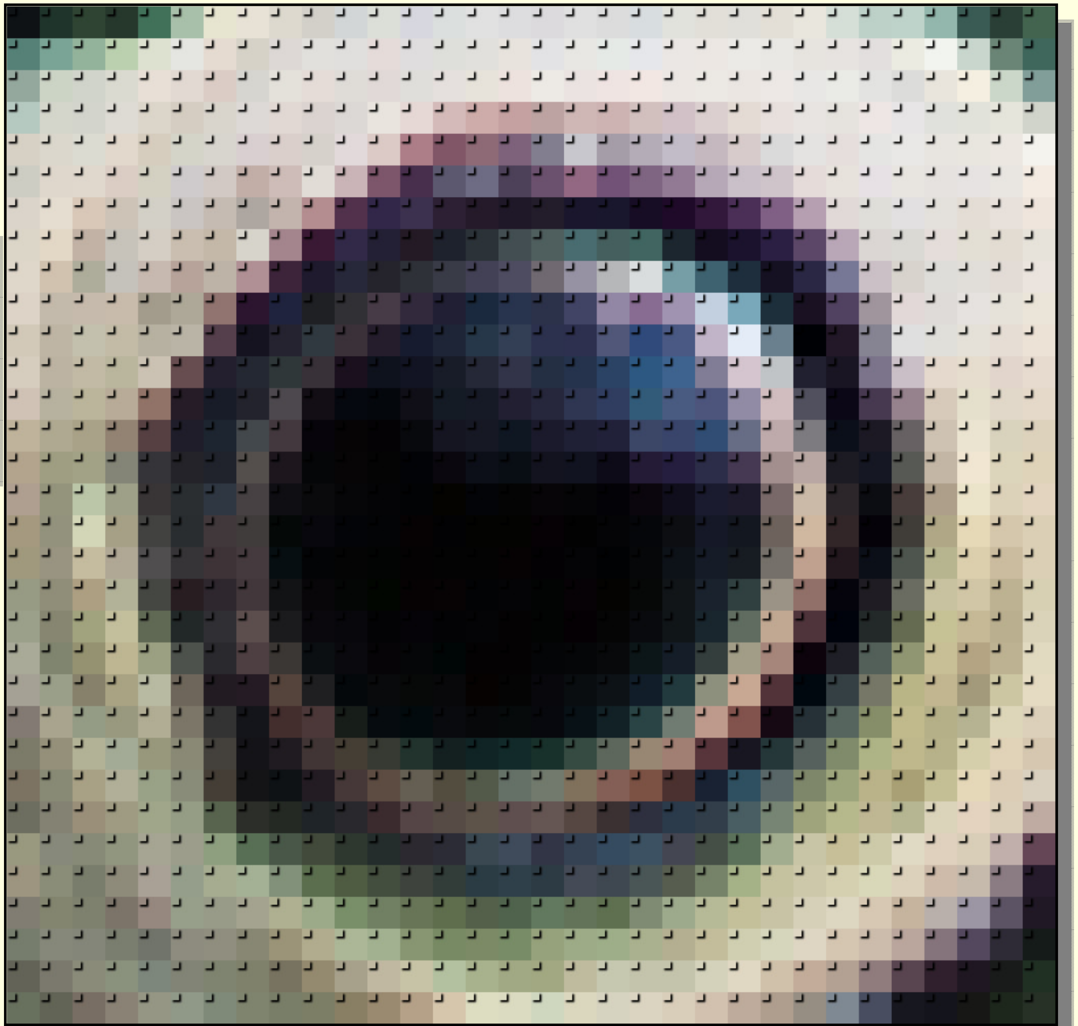
всеки пиксел се
повтаря в област 4×4



Билинейна интерполация - пример

Мащабиране: увеличаване 4×

Всеки 4-ти пиксел
на всеки 4-ти ред има
стойността на пиксел от
оригиналното изображение

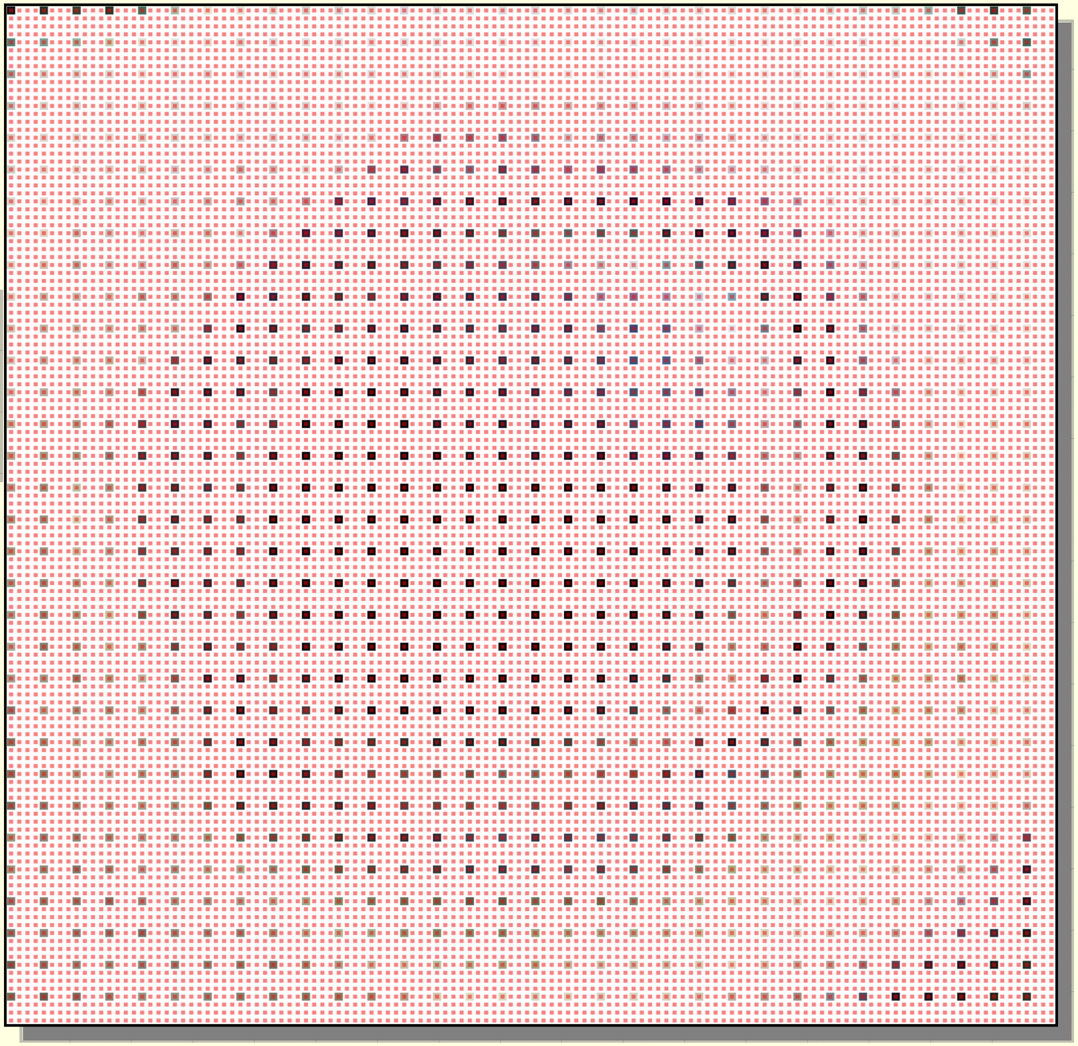


повторения +
оригинални стойности

Билинейна интерполация - пример

Мащабиране:
увеличаване 4×

За всяка област 4×4 трябва да се определят стойностите на останалите 15 пиксела



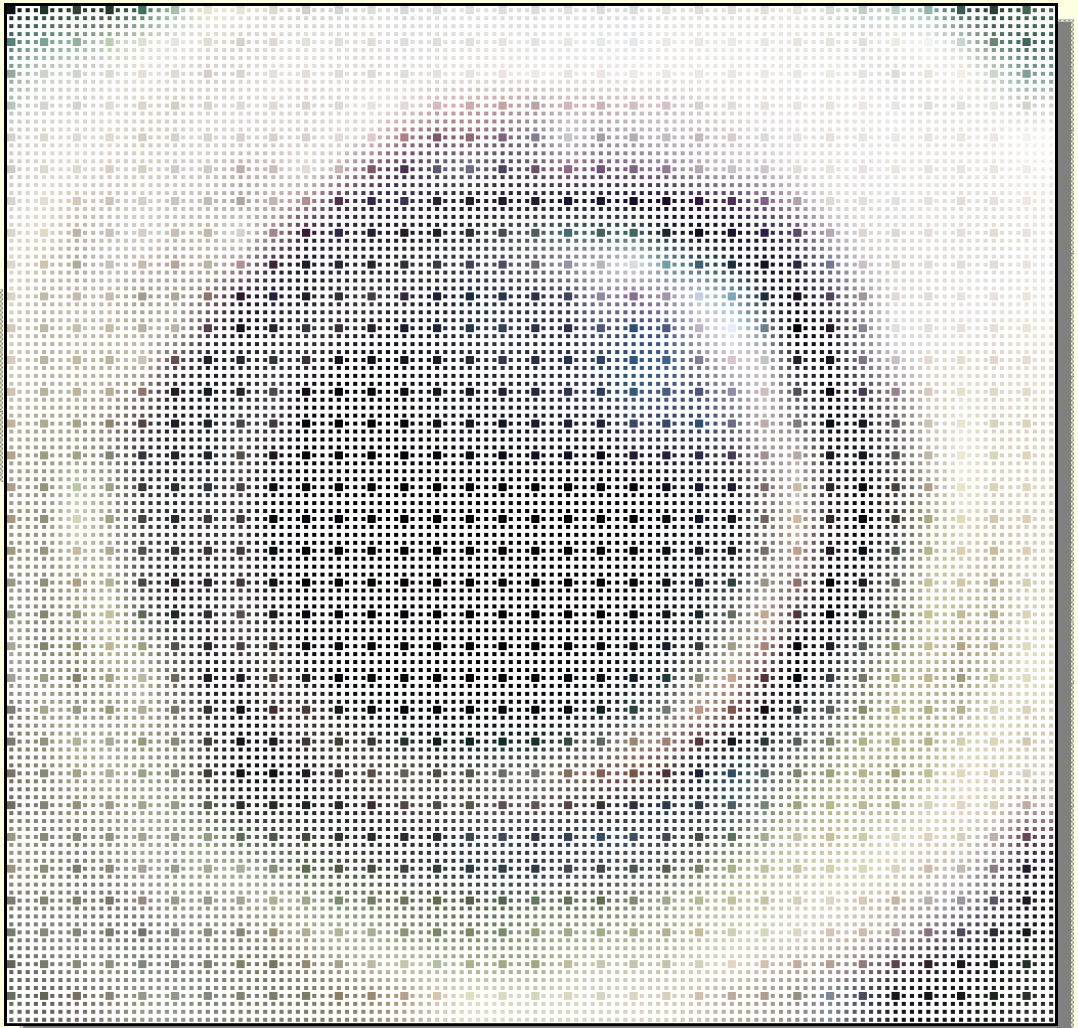
всяка розова точка е пиксел в изходното изображение

Билинейна интерполация - пример

Мащабиране:
увеличаване 4×

За всяка област 4×4 се определят междинни стойности

определят се междинни стойности за всички 15 пиксела



Билинейна интерполация - пример

Мащабиране: увеличаване 4×

Междинните стойности се попълват в изображението

всяка розова точка е пиксел в изходното изображение

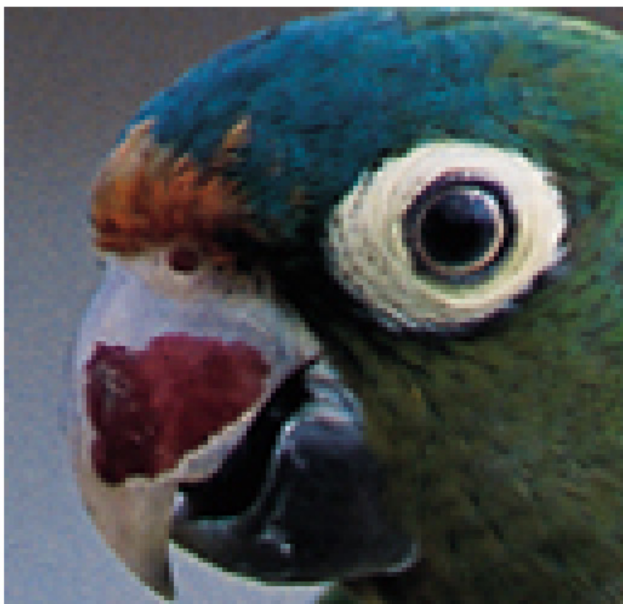
Билинейна интерполация - пример

Мащабиране:
увеличаване 4×

Краен резултат:
билинейна интерполация

Билинейна интерполация - пример

Мащабиране: увеличаване 4×

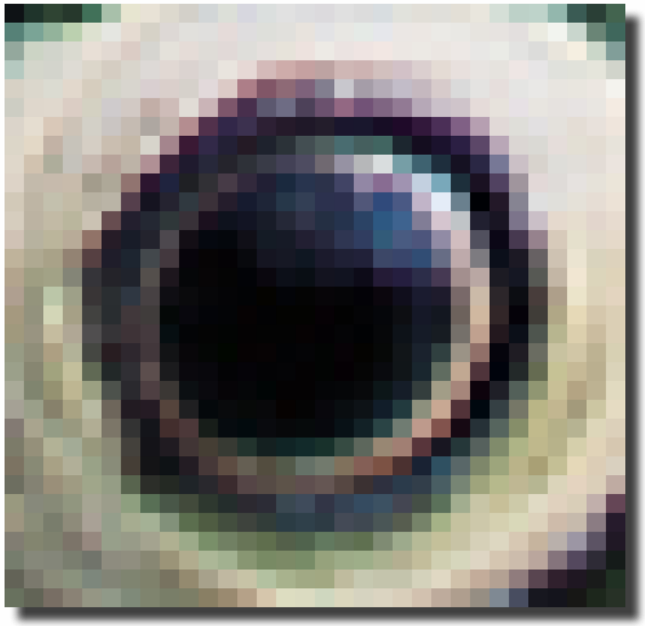


Най-близки съседи

Билинейна интерполация

Билинейна интерполация - пример

Мащабиране:
увеличаване 4×



Най-близки съседи

Билинейна интерполация

Бикубична интерполация

- Билинейна интерполация
  - стойностите на пикселите в изходното изображение се определят като претеглена сума от стойностите на 4-те съседни входни пиксела $I(r',c')$, $I(r'+1,c')$, $I(r'+1,c'+1)$ и $I(r',c'+1)$
- ***Бикубична интерполация***
  - освен тези 4 пиксела се използват и техните ***частни производни***

$$\left\{\left\{\frac{\partial}{\partial c'}\mathrm{I}(r'+i,c'+j),\ \frac{\partial}{\partial r'}\mathrm{I}(r'+i,c'+j),\ \frac{\partial}{\partial c'r'}\mathrm{I}(r'+i,c'+j)\right\}_{j=0}^{1}\right\}_{i=0}^{1}$$

$(r'\text{-}1,c'\text{-}1)$	$(r'\text{-}1,c')$	$(r'\text{-}1,c'+1)$	$(r'\text{-}1,c'+2)$
$(r',c'\text{-}1)$	(r',c')	$(r',c'+1)$	$(r',c'+2)$
$(r'+1,c'\text{-}1)$	$(r'+1,c')$	$(r'+1,c'+1)$	$(r'+1,c'+2)$
$(r'+2,c'\text{-}1)$	$(r'+2,c')$	$(r'+2,c'+1)$	$(r'+2,c'+2)$

Бикубична интерполация

- ***Бикубична интерполация***
  - производните се изчисляват за локални области на 4те съседни пиксела
  - за всеки пиксел в изходното изображение се разглежда ***локална околност 4×4***

$$\left\{\left\{\frac{\partial}{\partial c'}\mathrm{I}(r'+i,c'+j),\ \frac{\partial}{\partial r'}\mathrm{I}(r'+i,c'+j),\ \frac{\partial}{\partial c' r'}\mathrm{I}(r'+i,c'+j)\right\}_{j=0}^{1}\right\}_{i=0}^{1}$$

$(r'$-$1,c'$-$1)$	$(r'$-$1,c')$	$(r'$-$1,c'+1)$	$(r'$-$1,c'+2)$
$(r',c'$-$1)$	(r',c')	$(r',c'+1)$	$(r',c'+2)$
$(r'+1,c'$-$1)$	$(r'+1,c')$	$(r'+1,c'+1)$	$(r'+1,c'+2)$
$(r'+2,c'$-$1)$	$(r'+2,c')$	$(r'+2,c'+1)$	$(r'+2,c'+2)$

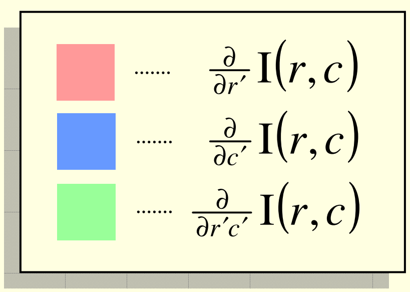
Бикубична интерполация

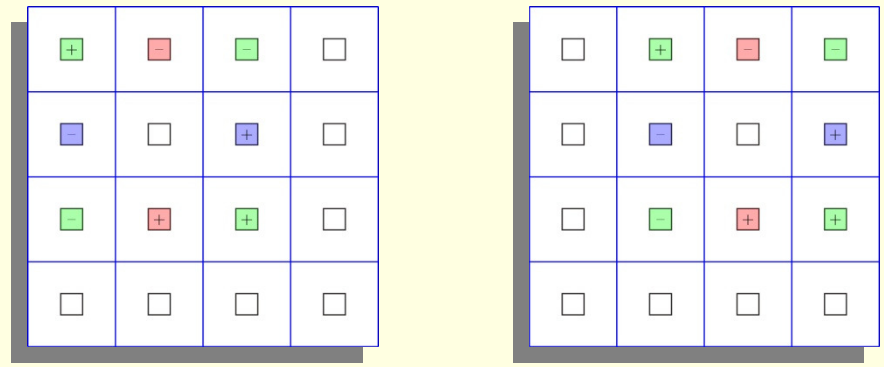
N(*r*',*c*') е локална област за пиксел (*r*',*c*') съдържаща 8 съседни пиксела

$\frac{\partial}{\partial r'}\mathrm{I}(r',c')=$ средна стойност на крайни разлики за (r',c')

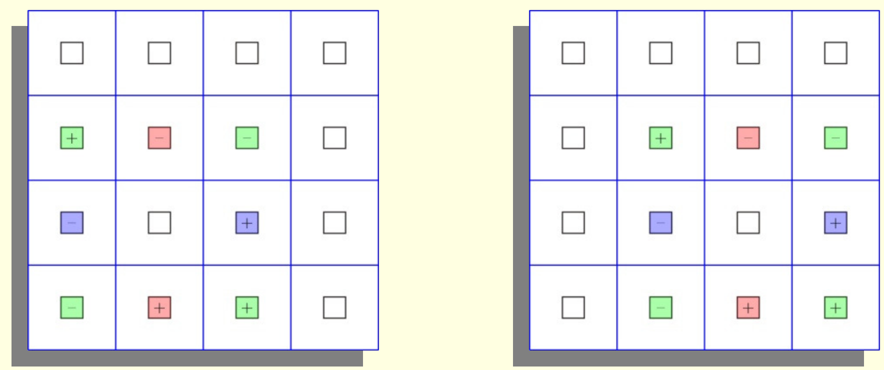
$$= \tfrac{1}{2}\left[\left(\mathrm{I}(r'+1,c')-\mathrm{I}(r',c')\right) + \left(\mathrm{I}(r',c')-\mathrm{I}(r'-1,c')\right)\right]$$

$$= \tfrac{1}{2}\left[\mathrm{I}(r'+1,c')-\mathrm{I}(r'-1,c')\right]$$





N (r',c') N $(r',c'+1)$



N $(r'+1,c')$ N $(r'+1,c'+1)$

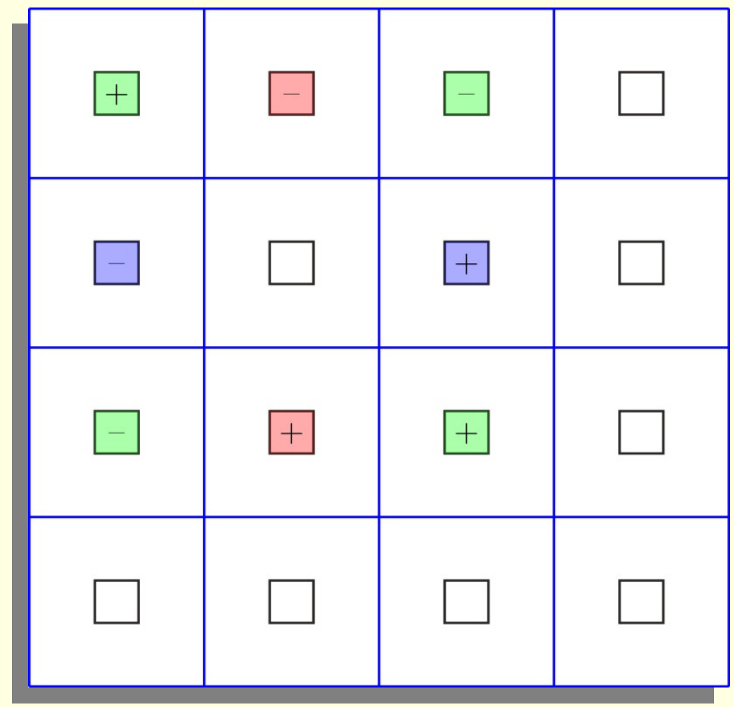
Бикубична интерполация

$$\mathrm{N}\left(r', c'\right)$$

$$\frac{\partial}{\partial r'} \mathrm{I}\left(r', c'\right) = \frac{1}{2}\left[\mathrm{I}\left(r'+1, c'\right) - \mathrm{I}\left(r'-1, c'\right)\right],$$

$$\frac{\partial}{\partial c'} \mathrm{I}\left(r', c'\right) = \frac{1}{2}\left[\mathrm{I}\left(r', c'+1\right) - \mathrm{I}\left(r', c'-1\right)\right],$$

$$\frac{\partial}{\partial c' r'} \mathrm{I}\left(r', c'\right) = \frac{1}{4}\Big[\mathrm{I}\left(r'+1, c'+1\right) - \mathrm{I}\left(r'+1, c'-1\right) + \mathrm{I}\left(r'-1, c'-1\right) - \mathrm{I}\left(r'-1, c'+1\right)\Big]$$



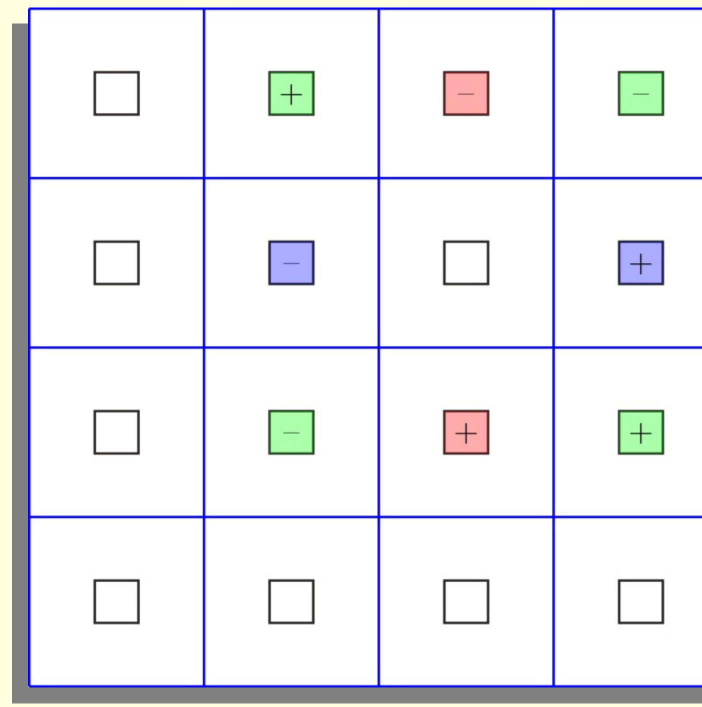
Бикубична интерполация

$$\mathrm{N}\,(r', c'+1)$$

$$\tfrac{\partial}{\partial r'}\mathrm{I}(r', c'+1) = \tfrac{1}{2}\left[\mathrm{I}(r'+1, c'+1) - \mathrm{I}(r'-1, c'+1)\right],$$

$$\tfrac{\partial}{\partial c'}\mathrm{I}(r', c'+1) = \tfrac{1}{2}\left[\mathrm{I}(r', c'+2) - \mathrm{I}(r', c')\right],$$

$$\begin{aligned}\tfrac{\partial}{\partial c' r'}\mathrm{I}(r', c'+1) = \tfrac{1}{4}\big[&\mathrm{I}(r'+1, c'+2) - \mathrm{I}(r'+1, c') \\ &+ \mathrm{I}(r'-1, c') - \mathrm{I}(r'-1, c'+2)\big]\end{aligned}$$



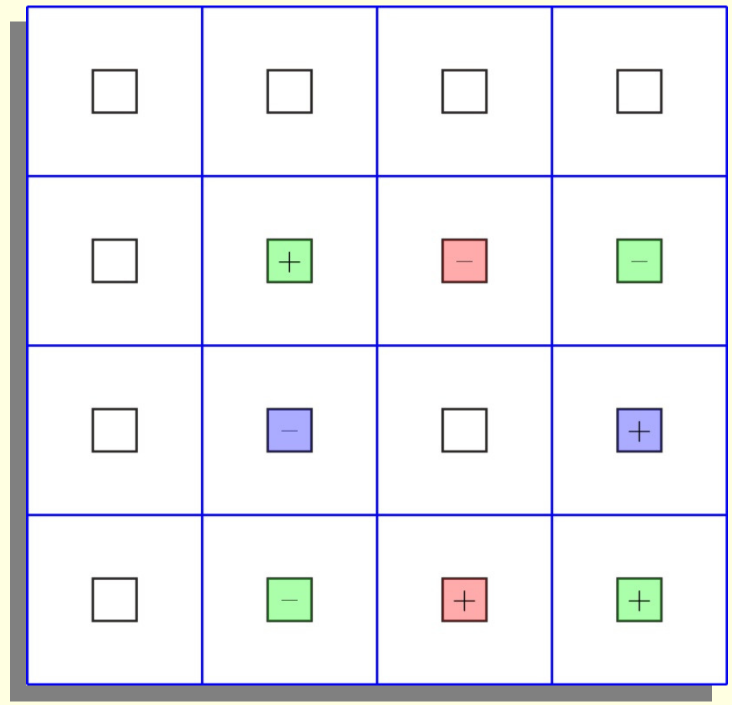
Бикубична интерполация

$$\mathrm{N}\left(r'+1, c'+1\right)$$

$$\tfrac{\partial}{\partial r'}\mathrm{I}\left(r'+1, c'+1\right) = \tfrac{1}{2}\left[\mathrm{I}\left(r'+2, c'+1\right) - \mathrm{I}\left(r', c'+1\right)\right],$$

$$\tfrac{\partial}{\partial c'}\mathrm{I}\left(r'+1, c'+1\right) = \tfrac{1}{2}\left[\mathrm{I}\left(r'+1, c'+2\right) - \mathrm{I}\left(r'+1, c'\right)\right],$$

$$\tfrac{\partial}{\partial c' r'}\mathrm{I}\left(r'+1, c'+1\right) = \tfrac{1}{4}\left[\mathrm{I}\left(r'+2, c'+2\right) - \mathrm{I}\left(r'+2, c'\right) + \mathrm{I}\left(r', c'\right) - \mathrm{I}\left(r', c'+2\right)\right]$$



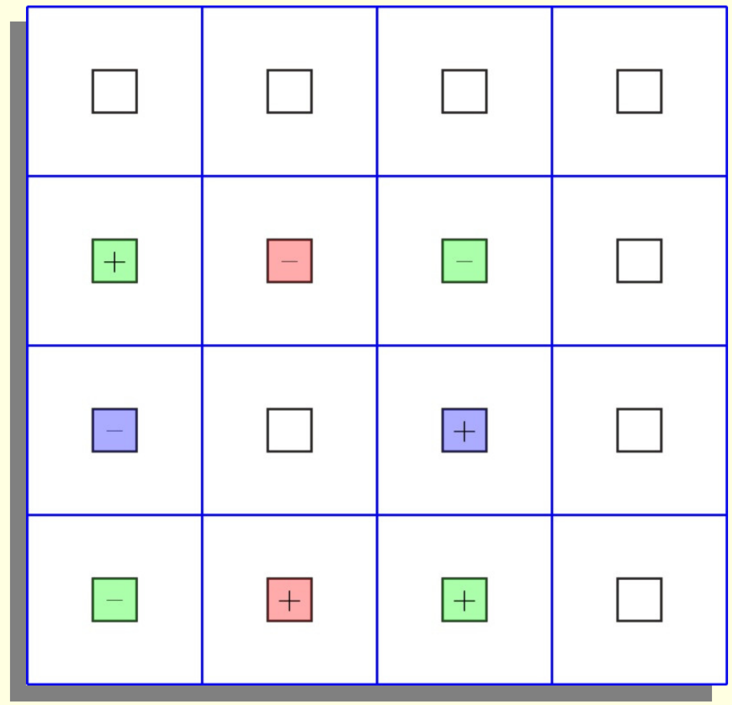
Бикубична интерполация

$$N\left(r'+1, c'\right)$$

$$\tfrac{\partial}{\partial r'} \mathrm{I}\left(r'+1, c'\right) = \tfrac{1}{2}\left[\mathrm{I}\left(r'+2, c'\right) - \mathrm{I}\left(r', c'\right)\right],$$

$$\tfrac{\partial}{\partial c'} \mathrm{I}\left(r'+1, c'\right) = \tfrac{1}{2}\left[\mathrm{I}\left(r'+1, c'+1\right) - \mathrm{I}\left(r'+1, c'-1\right)\right],$$

$$\begin{aligned}\tfrac{\partial}{\partial c' r'} \mathrm{I}\left(r'+1, c'\right) = \tfrac{1}{4}\Big[&\mathrm{I}\left(r'+2, c'+1\right) - \mathrm{I}\left(r'+2, c'-1\right) \\ &+ \mathrm{I}\left(r', c'-1\right) - \mathrm{I}\left(r', c'+1\right)\Big]\end{aligned}$$



Бикубична интерполация

$$S_r = \begin{cases} R'/R, & \text{if} \quad R' > R, \\ (R'-1)/R, & \text{if} \quad R' < R. \end{cases}$$

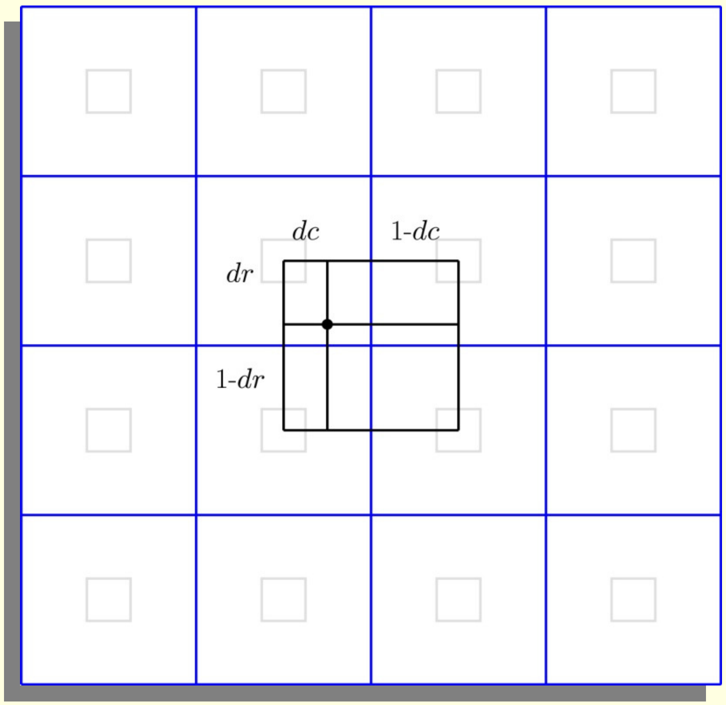
$$S_c = \begin{cases} C'/C, & \text{if} \quad C' > C, \\ (C'-1)/C, & \text{if} \quad C' < C. \end{cases}$$

$$r_f = [(1,\ldots,R)\cdot S_r], c_f = [(1,\ldots,C)\cdot S_c]$$

$$(r', c') = (\lfloor r_f \rfloor, \lfloor c_f \rfloor)$$

$$(dr, dc) = (r_f - r', c_f - c')$$

Аналогично на интерполация по най-близки съседи

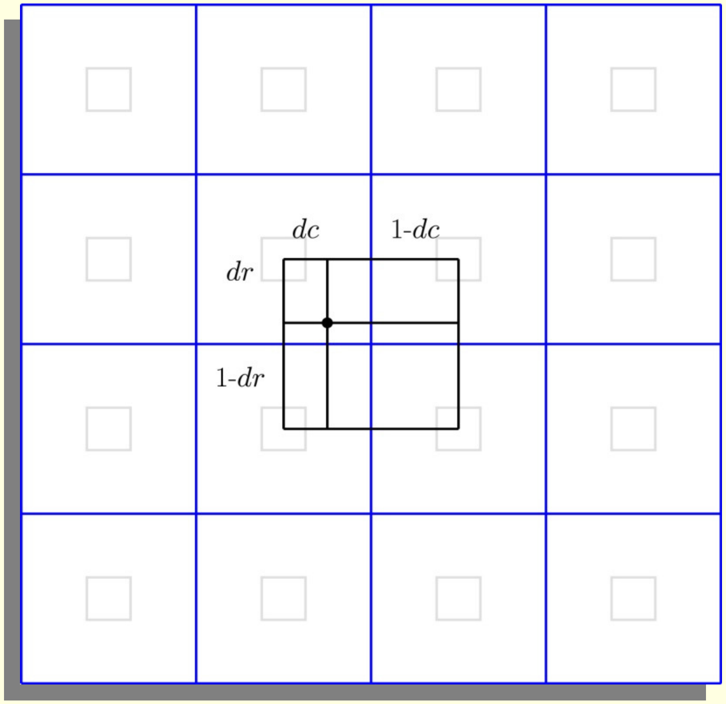


Бикубична интерполация

$$J(r,c)=\sum_{m=-1}^{2}\sum_{n=-1}^{2} I(r'+m,c'+n)P(dr-m)P(n-dc)$$

$$P(x)=\tfrac{1}{6}\left[Q(x+2)^3-4Q(x+1)^3-6Q(x)^3-4Q(x-1)^3\right]$$

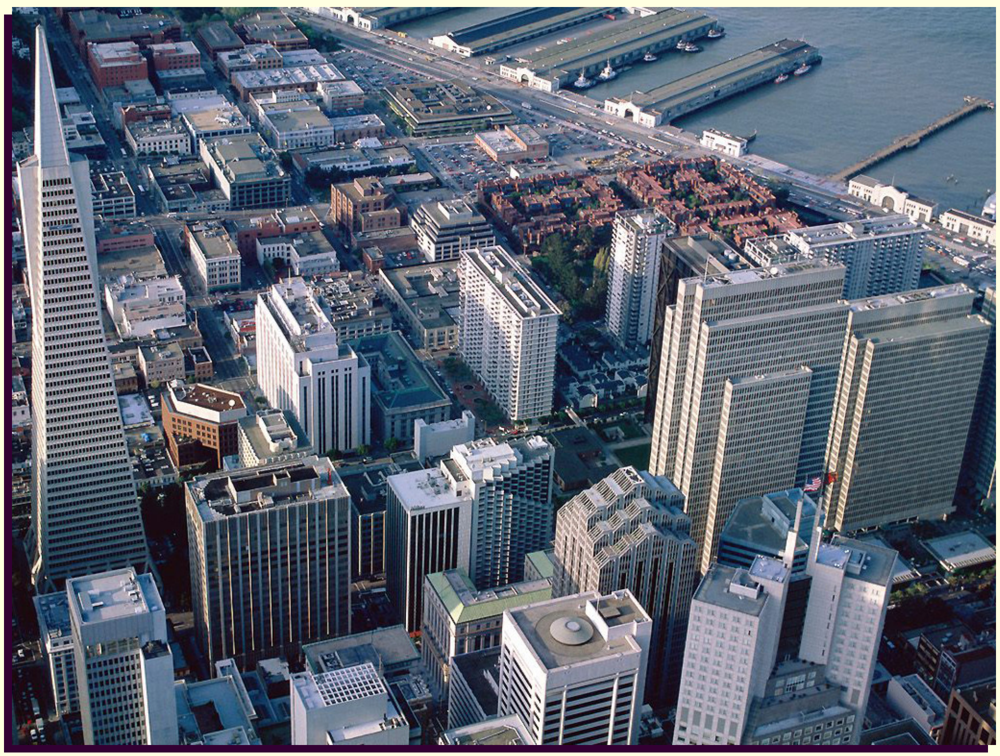
$$Q(x)=\begin{cases} x \text{ за } x>0 \\ 0 \text{ за } x\le 0 \end{cases}$$



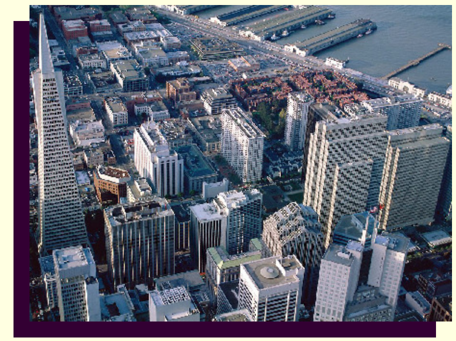
Производните за всеки от 4те съседни пиксела са комбинирани в сума на произведения на полиноми

Бикубична интерполация

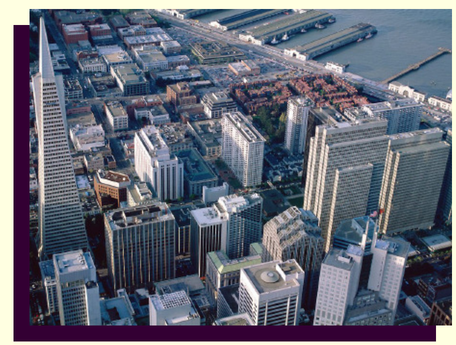
Пример: мащабиране на изображение до 3/7 от размерите на оригиналното изображение



оригинално изображение



най-близки съседи



билинейна инт.

бикубична инт.

Бикубична интерполация

Мащабиране: намаляване 3/7

оригинално изображение

Бикубична интерполация

Мащабиране: намаляване 3/7 (увеличено)

най-близки съседи

Бикубична интерполация

Мащабиране: намаляване 3/7 (увеличено)

билинейна интерполация

Бикубична интерполация

Мащабиране: намаляване 3/7 (увеличено)

бикубична интерполация

Бикубична интерполация

Мащабиране: увеличаване 7/3

най-близки съседи

Бикубична интерполация

Мащабиране: увеличаване 7/3

билинейна
интерполация

Бикубична интерполация

Мащабиране: увеличаване 7/3

бикубична интерполация

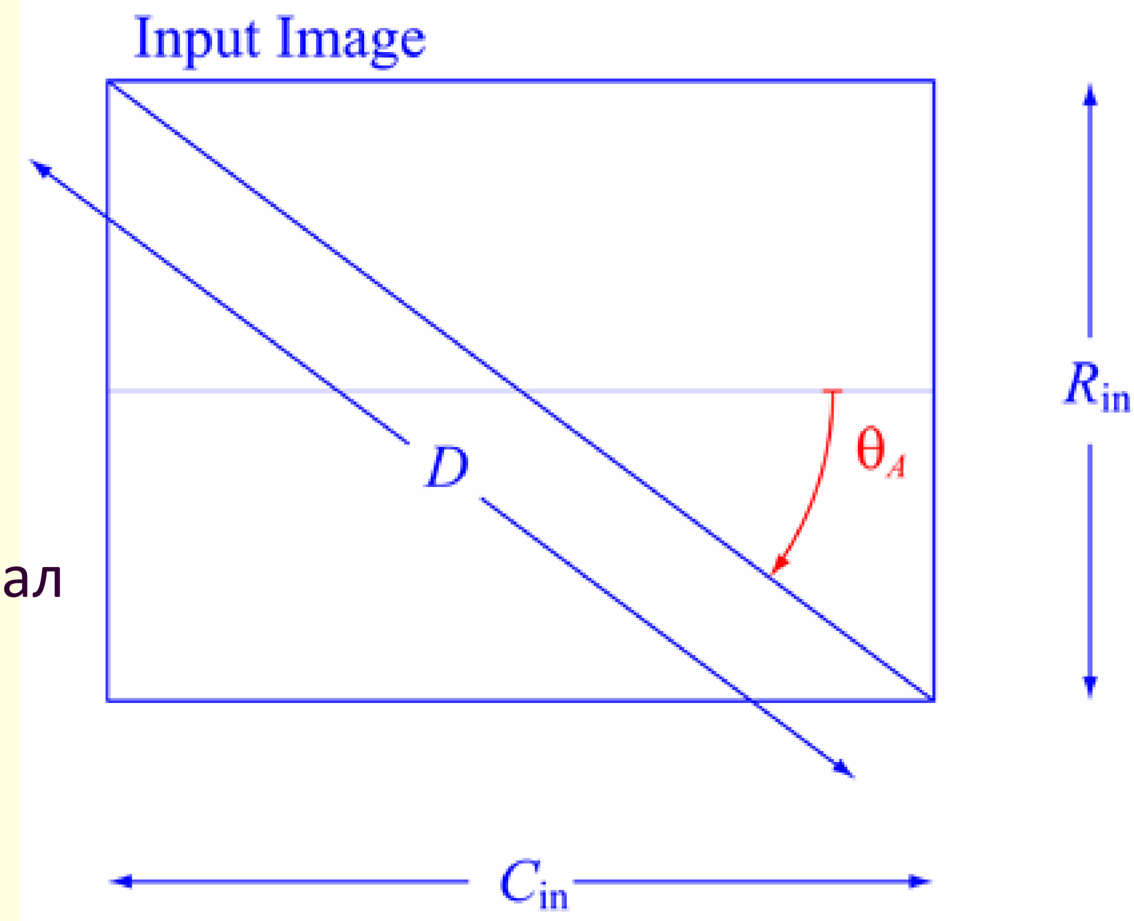
Геометрично изкривяване

- Входно изображение *I*
  - размери $R_{in} \times C_{in}$
- Ъгъл на завъртане

$$\theta_A = \tan^{-1}\left[\frac{R_{in}}{C_{in}}\right]$$

- Дължина на диагонал

$$D = \sqrt{R_{in}^2 + C_{in}^2}$$

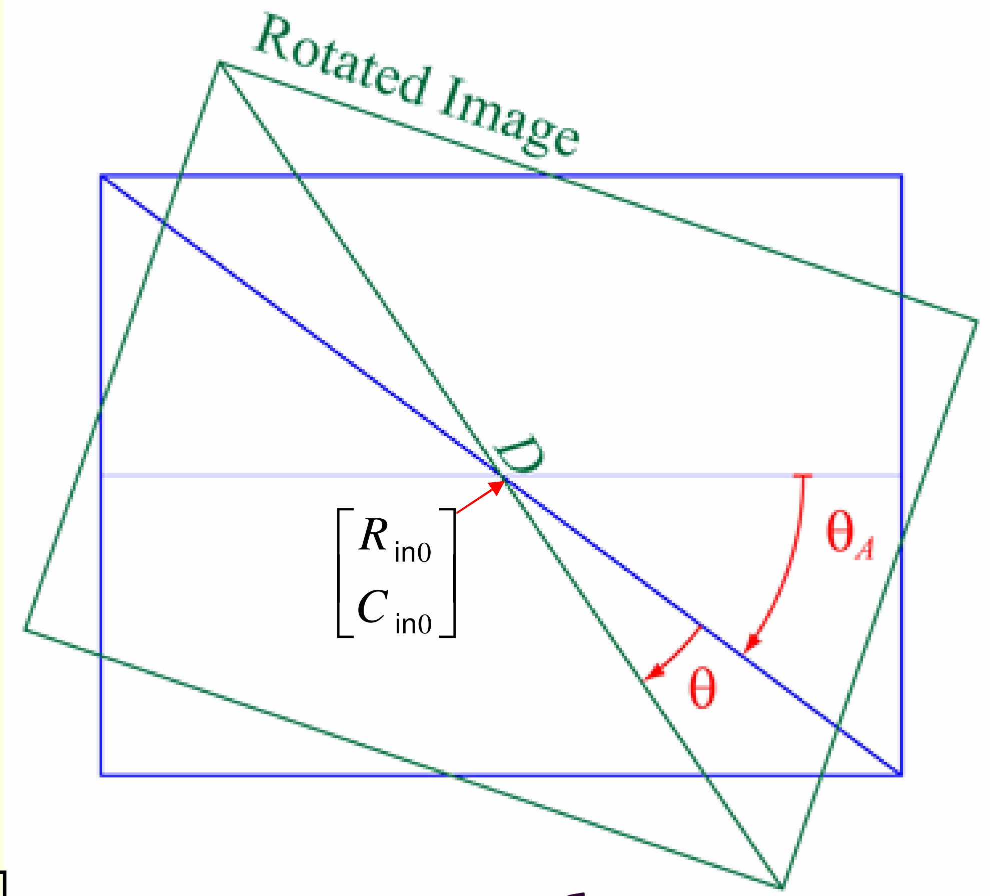


Геометрично изкривяване

- ъгъл на завъртане θ
- ротационна матрица

$$P(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

- трансформация на координати на входното изображение към координати в изходното изображение



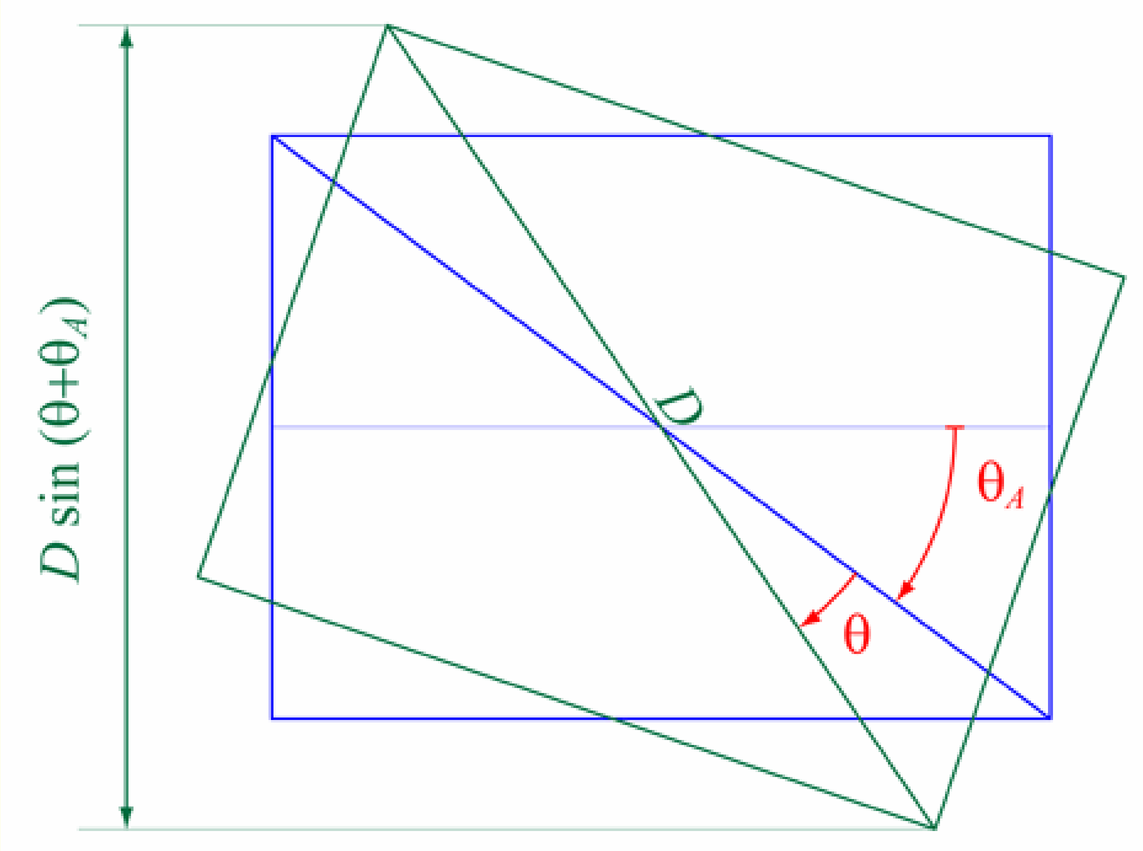
$$\begin{bmatrix} r \\ c \end{bmatrix} = P(\theta)\begin{bmatrix} r_{\text{in}} - R_{\text{in}0} \\ c_{\text{in}} - C_{\text{in}0} \end{bmatrix} + \begin{bmatrix} R_{\text{out}0} \\ R_{\text{out}0} \end{bmatrix}$$

- център на изображението

$$\left(R_{\text{in}0}, C_{\text{in}0}\right) = \left(\left\lfloor \tfrac{1}{2} R_{\text{in}} \right\rfloor + 1, \left\lfloor \tfrac{1}{2} C_{\text{in}} \right\rfloor + 1\right)$$

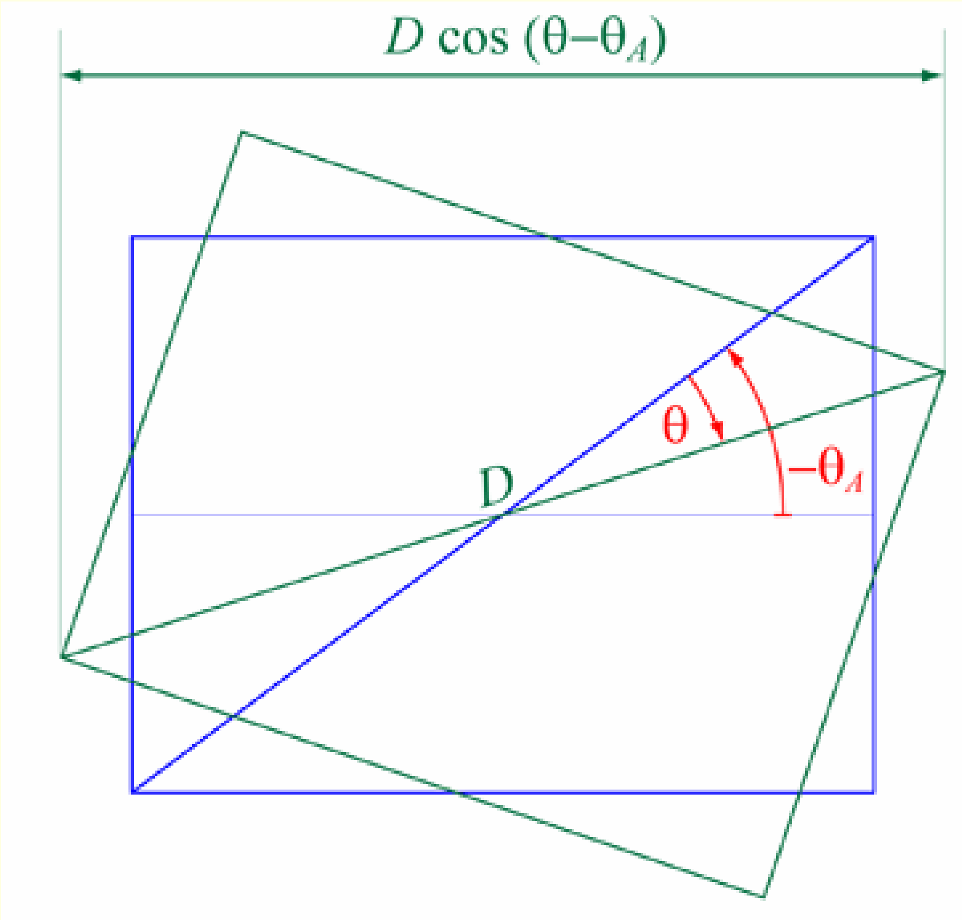
Геометрично изкривяване

- размери на изходното изображение
  - редове: $R_{\text{out}} = round\left(\left|\mathrm{D}\sin(\theta+\theta_A)\right|\right)$ ако $0° < \theta \leq 90°$



Геометрично изкривяване

- размери на изходното изображение
  - колони: $C_{\text{out}} = round\left(\left|\text{D}\cos(\theta - \theta_A)\right|\right)$ ако $0° < \theta \leq 90°$



Геометрично изкривяване

- размери на изходното изображение
  - зависят от стойността на ъгъла на завъртане

$0° \le \theta < 90°:$

$$R_{\text{out}} = round\left(\left|D\sin(\theta + \theta_A)\right|\right)$$

$$C_{\text{out}} = round\left(\left|D\cos(\theta - \theta_A)\right|\right)$$

$-90° \le \theta < 0°:$

$$R_{\text{out}} = round\left(\left|D\sin(\theta - \theta_A)\right|\right)$$

$$C_{\text{out}} = round\left(\left|D\cos(\theta + \theta_A)\right|\right)$$

$90° \le \theta < 180°:$

$$R_{\text{out}} = round\left(\left|D\cos(\theta - 90 - \theta_A)\right|\right)$$

$$C_{\text{out}} = round\left(\left|D\sin(\theta - 90 + \theta_A)\right|\right)$$

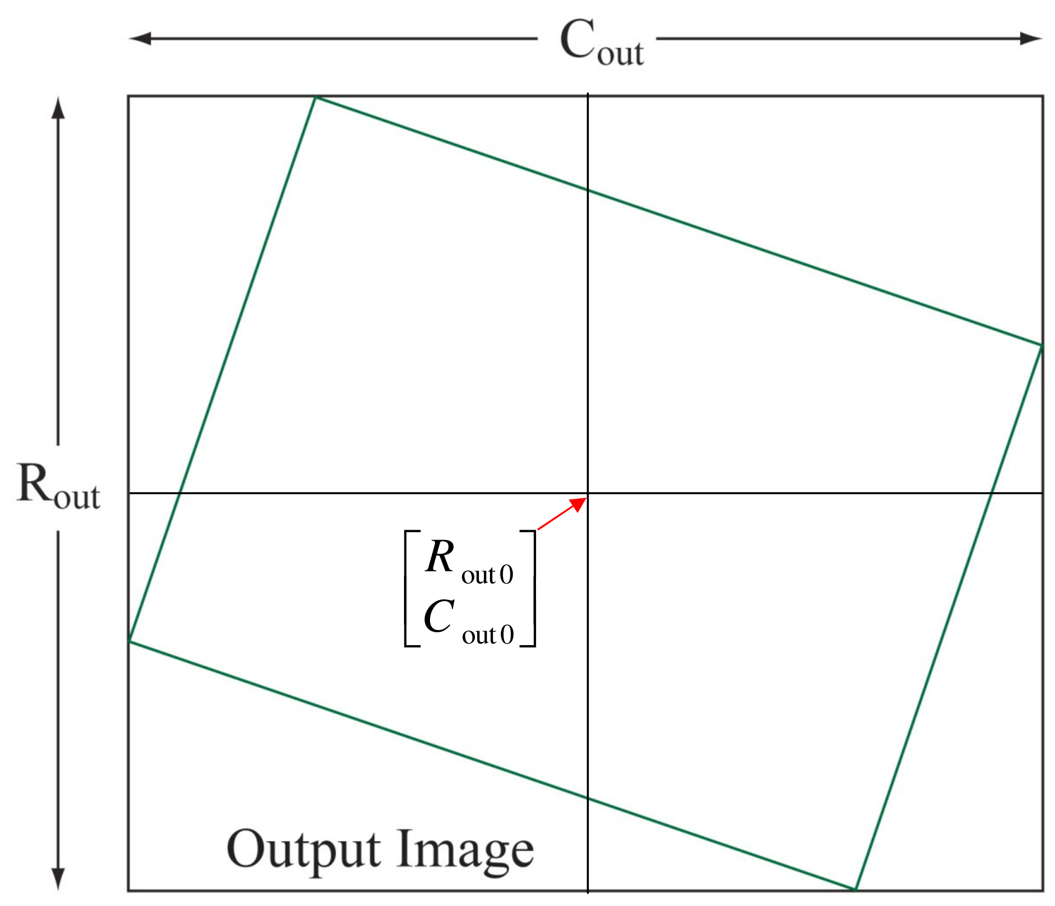
$-180° \le \theta < -90°:$

$$R_{\text{out}} = round\left(\left|D\cos(\theta + 90 + \theta_A)\right|\right)$$

$$C_{\text{out}} = round\left(\left|D\sin(\theta + 90 - \theta_A)\right|\right)$$

Геометрично изкривяване

- Определят се размерите на изходното изображение *J*
  - $R_{out} \times C_{out}$



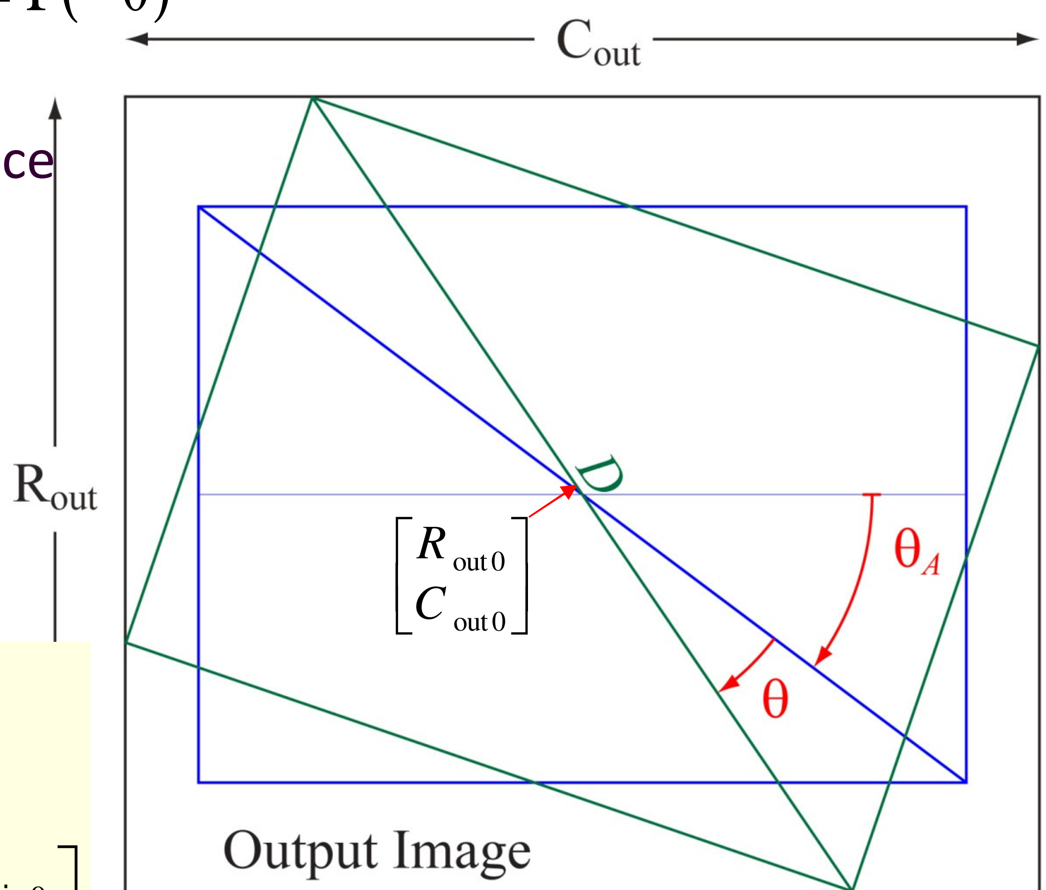
- Определя се централната точка

$$\left(R_{out0}, C_{out0}\right) = \left(\left\lfloor \tfrac{1}{2} R_{out} \right\rfloor + 1, \left\lfloor \tfrac{1}{2} C_{out} \right\rfloor + 1\right)$$

Геометрично изкривяване

$$P^{-1}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = P(-\theta)$$

- За всеки пиксел (r,c) се определя пиксел от входното изображение (r_f ,c_f) чрез ротация на (r,c) на ъгъл –θ около центъра (R_{out0},C_{out0})

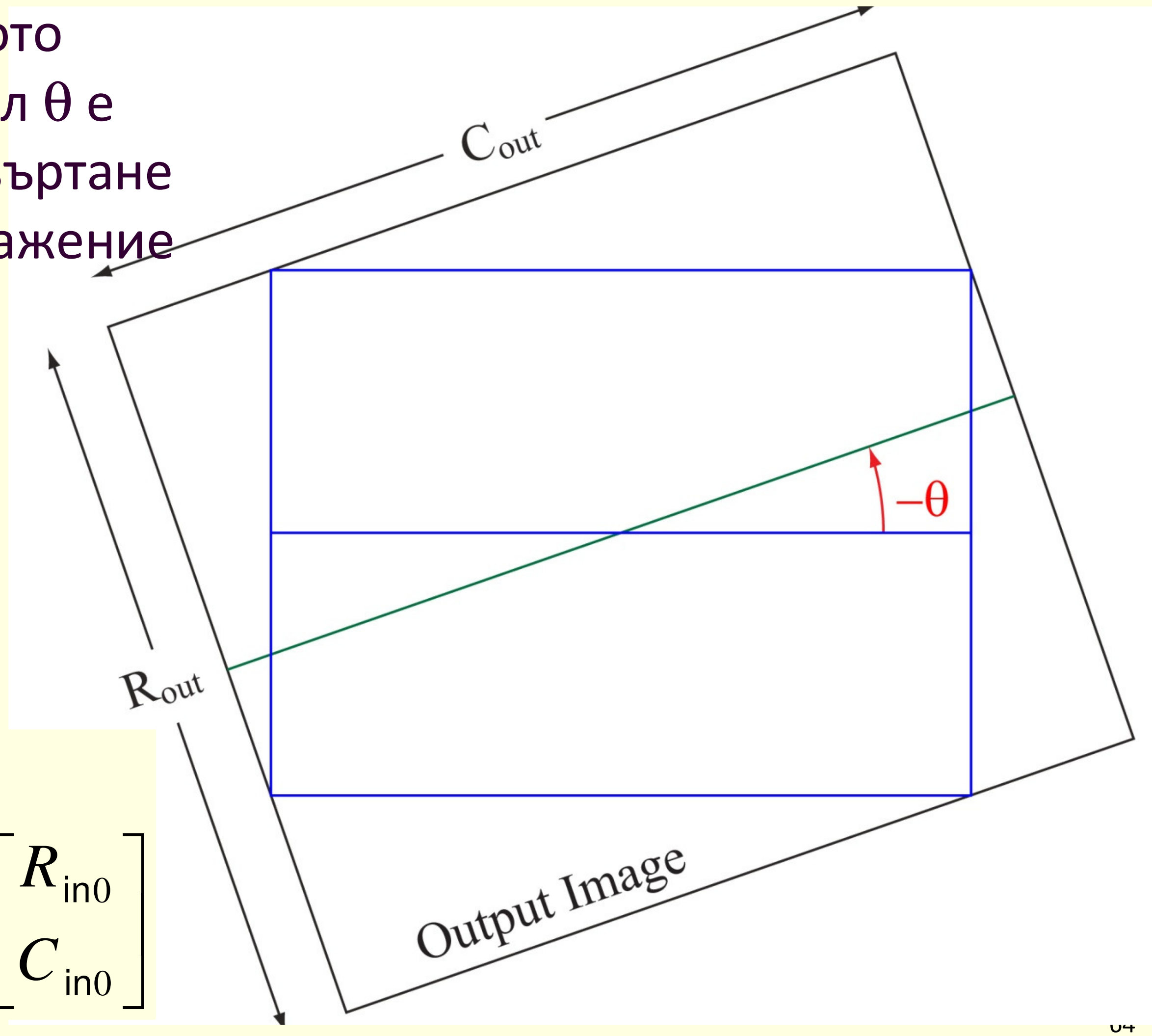


$$\Phi(r,c,:) = \begin{bmatrix} r_f \\ c_f \end{bmatrix} =$$

$$= P^{-1}(\theta)\begin{bmatrix} r - R_{out0} \\ c - C_{out0} \end{bmatrix} + \begin{bmatrix} R_{in0} \\ C_{in0} \end{bmatrix}$$

Геометрично изкривяване

- Ротацията на входното изображение на ъгъл θ е еквивалентна на завъртане на изходното изображение на ъгъл –θ

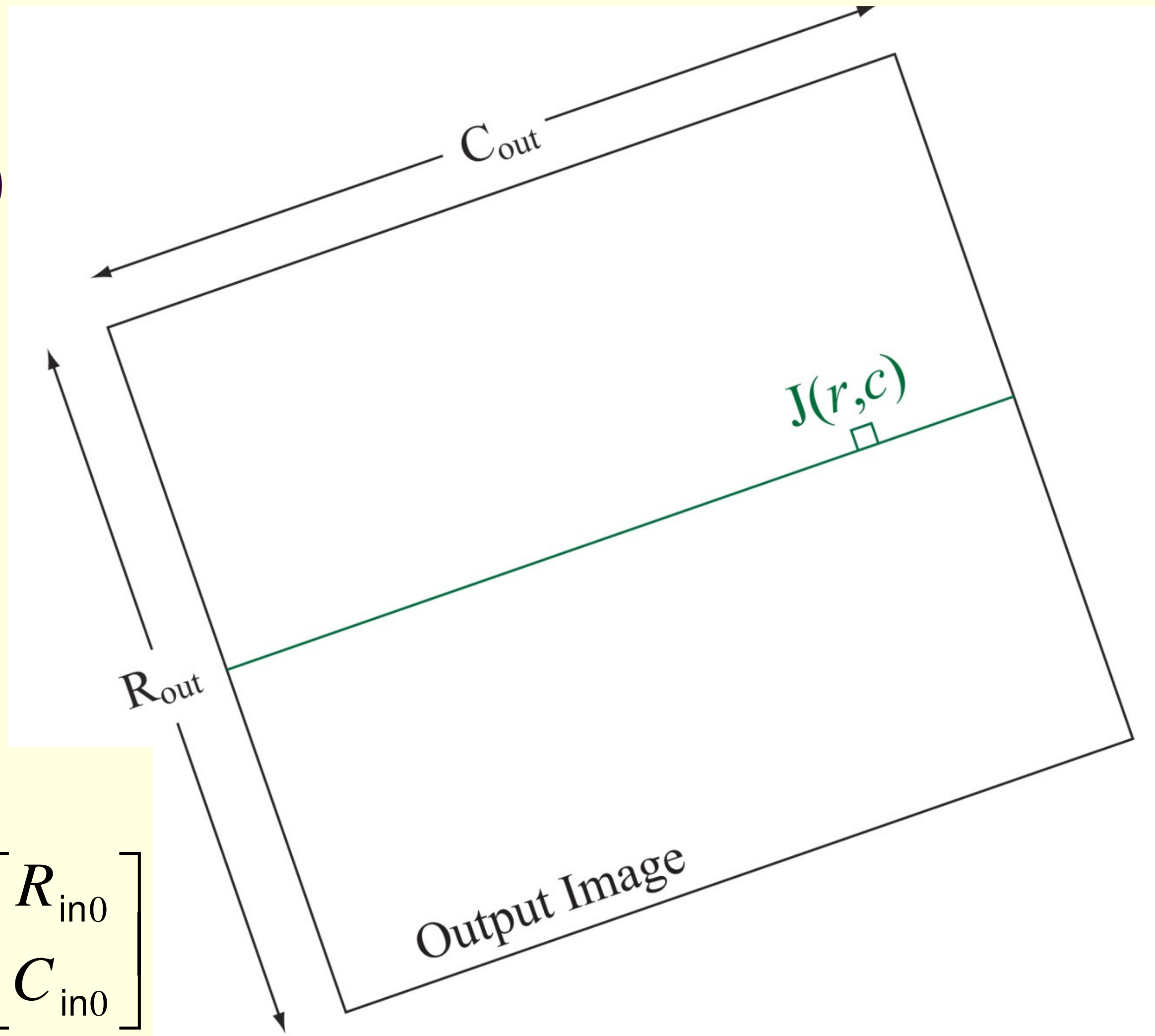


$$\Phi(r,c,:)=$$

$$= \mathrm{P}^{-1}(\theta)\begin{bmatrix} r - R_{\mathrm{out}0} \\ c - C_{\mathrm{out}0} \end{bmatrix} + \begin{bmatrix} R_{\mathrm{in}0} \\ C_{\mathrm{in}0} \end{bmatrix}$$

Геометрично изкривяване

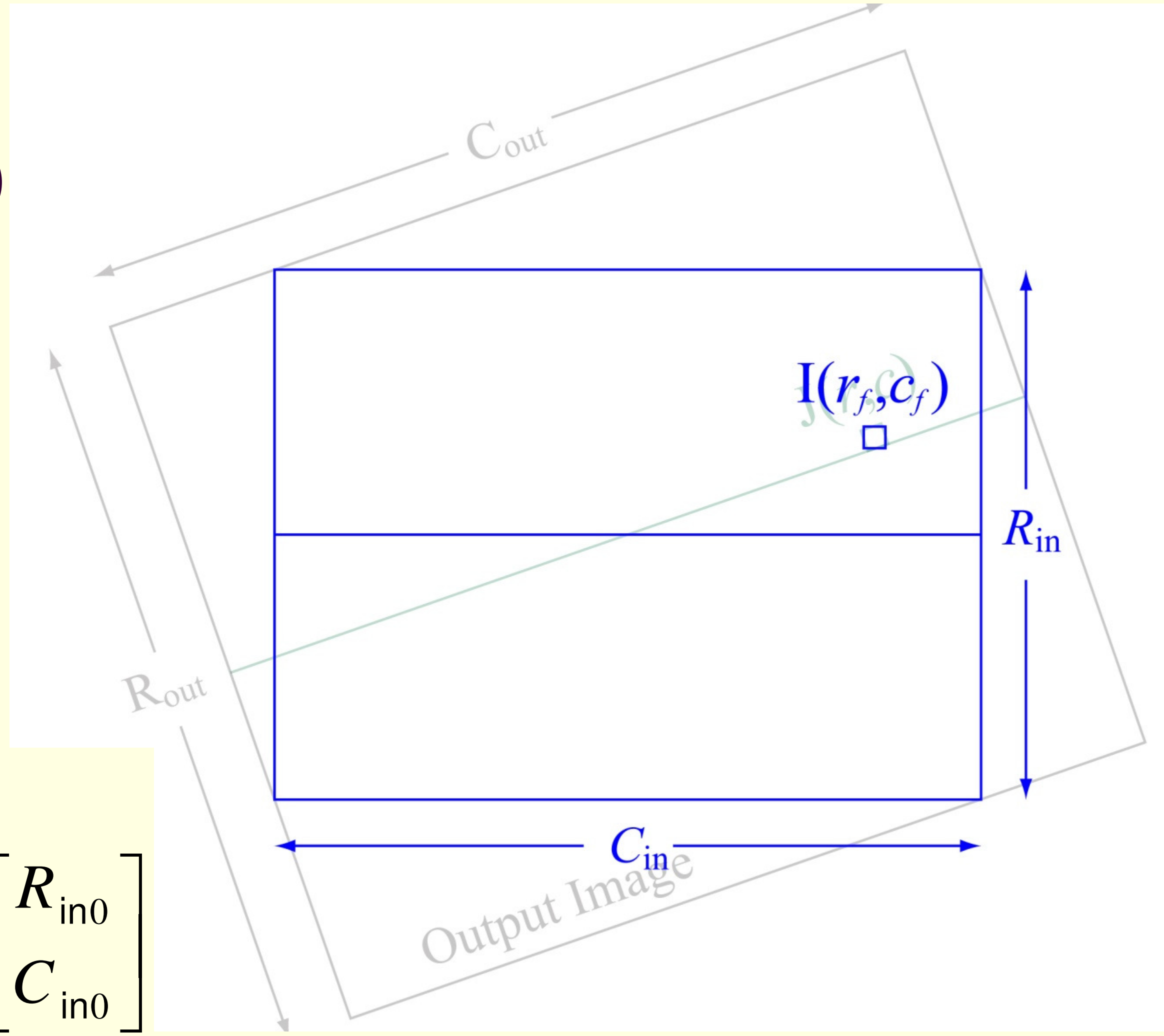
- След ротацията *J(r,c)* е в почти същата позиция както $I(r_f, c_f)$



$$\Phi(r,c,:) = \\ = \mathrm{P}^{-1}(\theta)\begin{bmatrix} r - R_{\mathrm{out}0} \\ c - C_{\mathrm{out}0} \end{bmatrix} + \begin{bmatrix} R_{\mathrm{in}0} \\ C_{\mathrm{in}0} \end{bmatrix}$$

Геометрично изкривяване

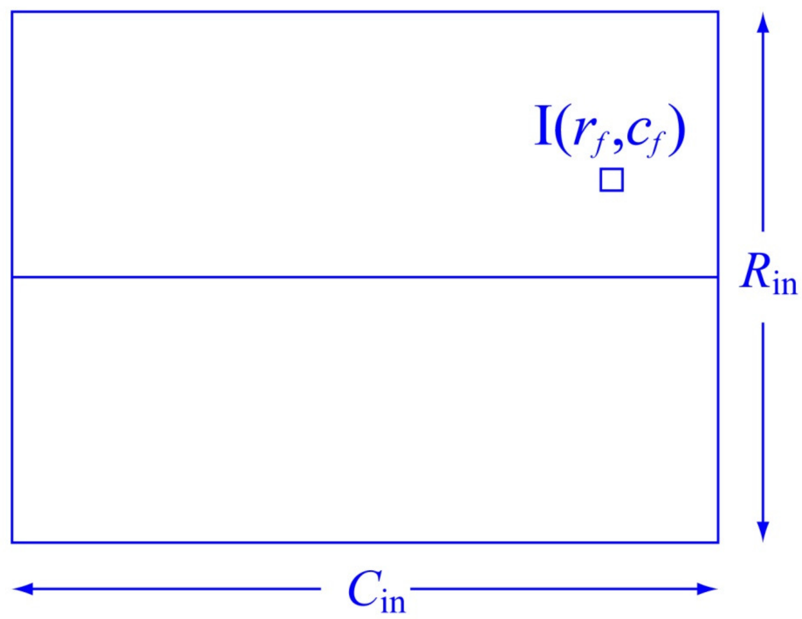
- След ротацията *J(r,c)* е в почти същата позиция както $I(r_f, c_f)$



$$\Phi(r,c,:) =$$
$$= P^{-1}(\theta)\begin{bmatrix} r - R_{\text{out}0} \\ c - C_{\text{out}0} \end{bmatrix} + \begin{bmatrix} R_{\text{in}0} \\ C_{\text{in}0} \end{bmatrix}$$

Геометрично изкривяване

- След ротацията $J(r,c)$ е в почти същата позиция както $I(r_f, c_f)$



$$\Phi(r,c,:) =$$

$$= \mathrm{P}^{-1}(\theta)\begin{bmatrix} r - R_{\text{out}0} \\ c - C_{\text{out}0} \end{bmatrix} + \begin{bmatrix} R_{\text{in}0} \\ C_{\text{in}0} \end{bmatrix}$$

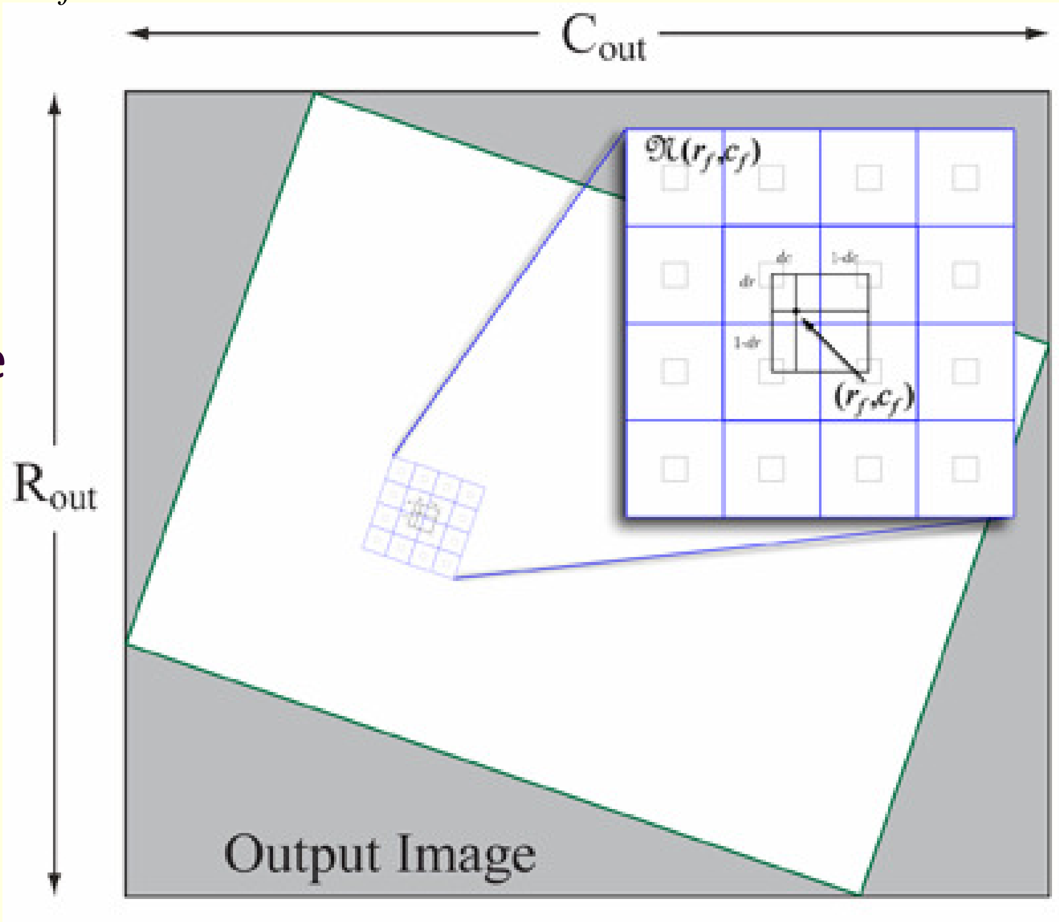
Геометрично изкривяване

$$J(r, c) = \Theta\{I; \mathbf{N}(r_f, c_f)\}.$$

Интерполация

- стойността в изходното изображение *J(r,c)* е функция на стойностите в локална област за пиксел (r_f ,c_f)

- *Най-близки съседи*
- *Билинейна*
- *Бикубична*

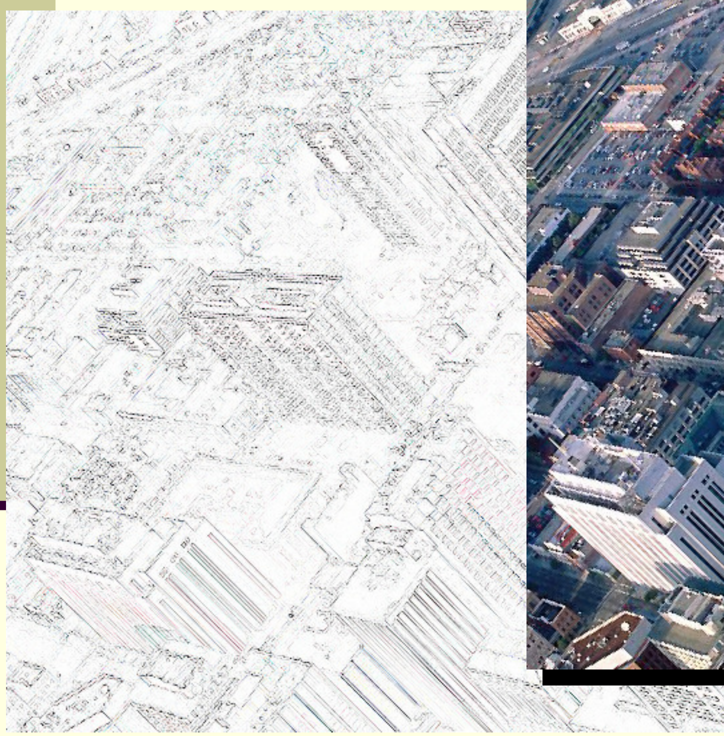


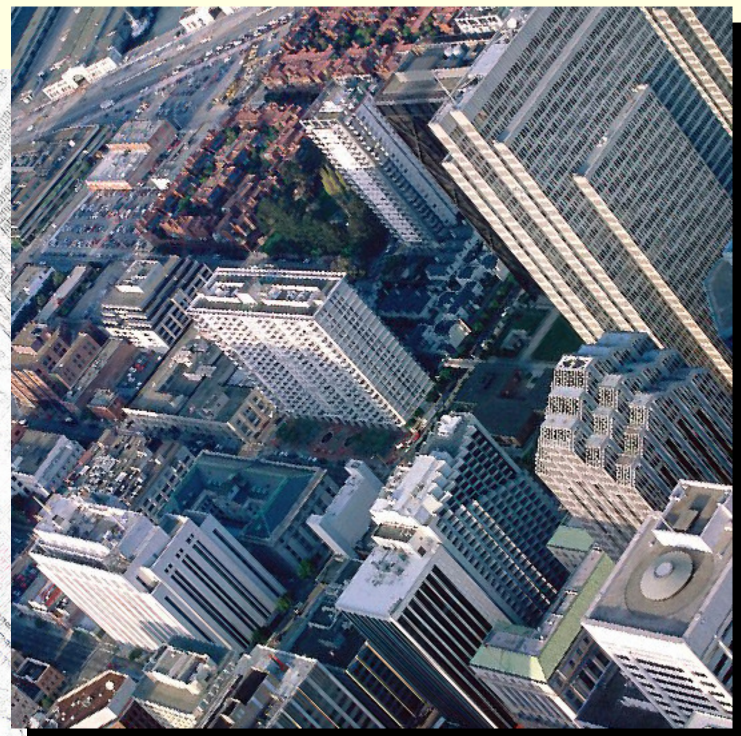
Геометрично изкривяване – пример

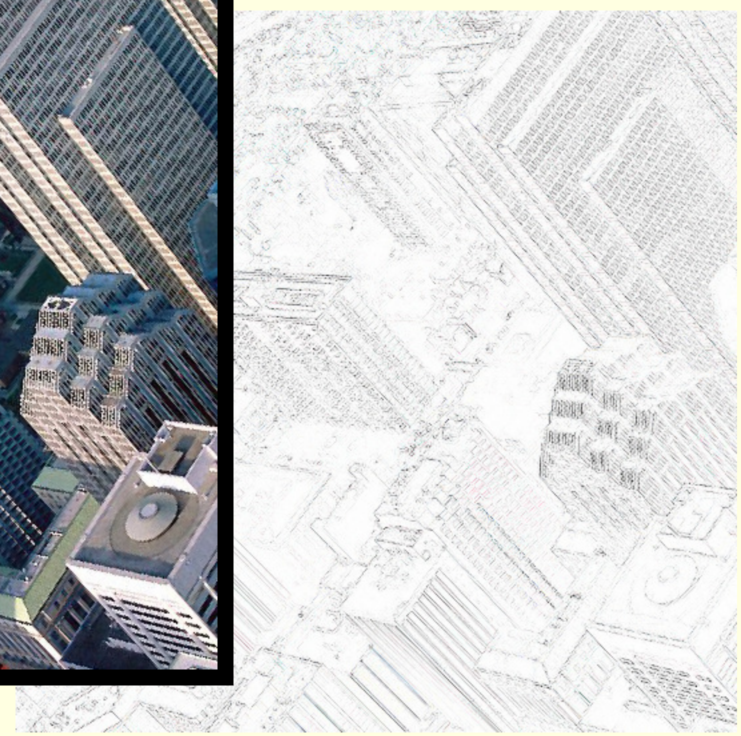
оригинално изображение

Геометрично изкривяване – пример

Ротация с най-близки съседи





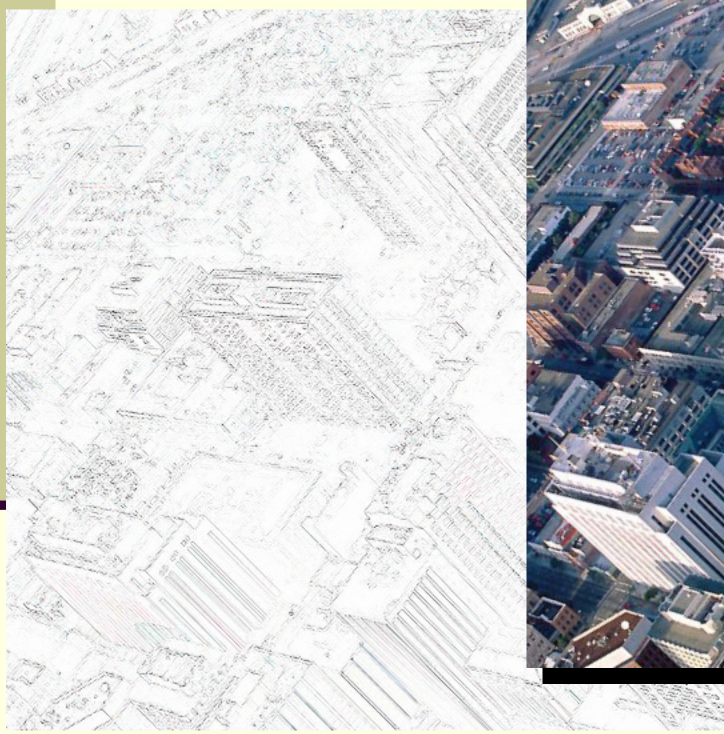


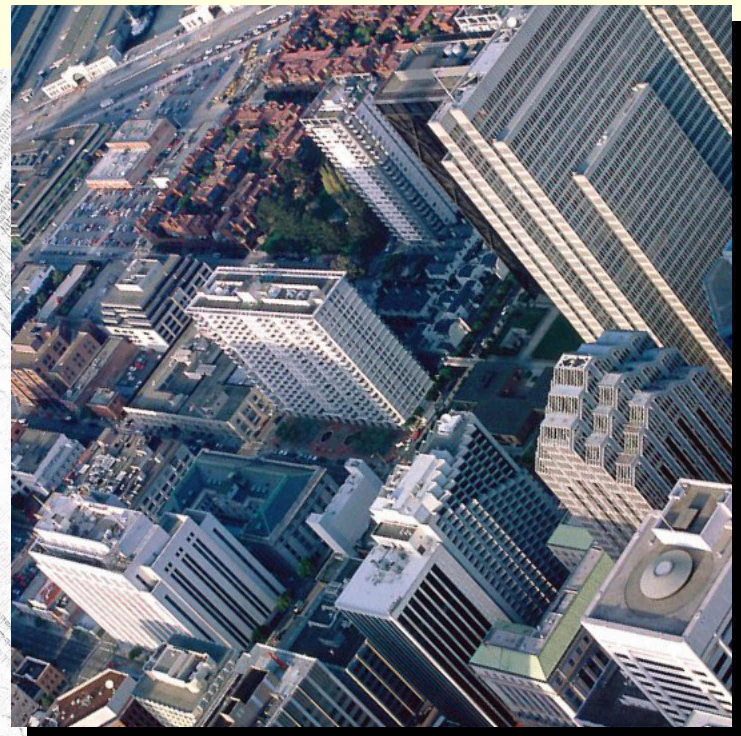
Bicubic – Nearest Neighbor

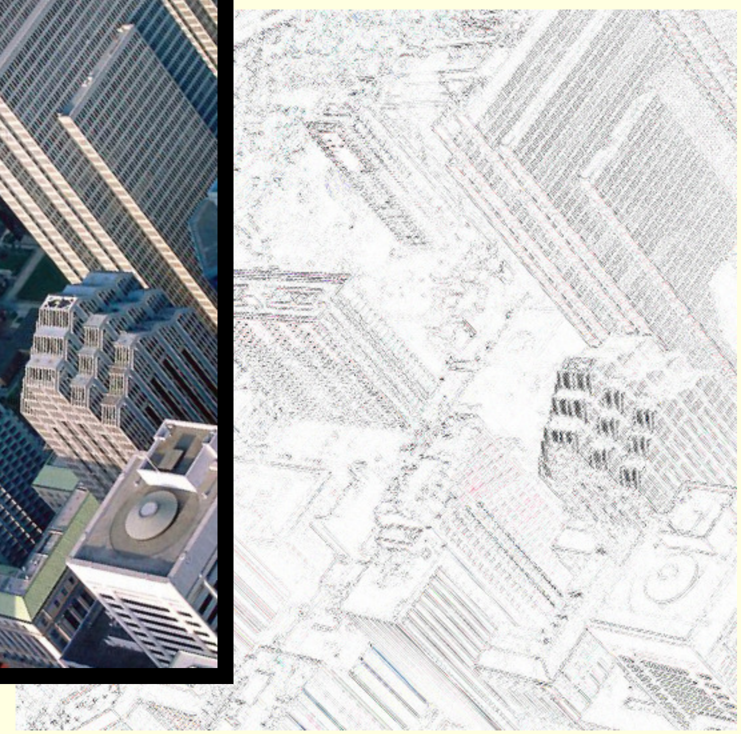
Nearest Neighbor – Bilinear

Геометрично изкривяване – пример

Ротация с билинейна интерполация





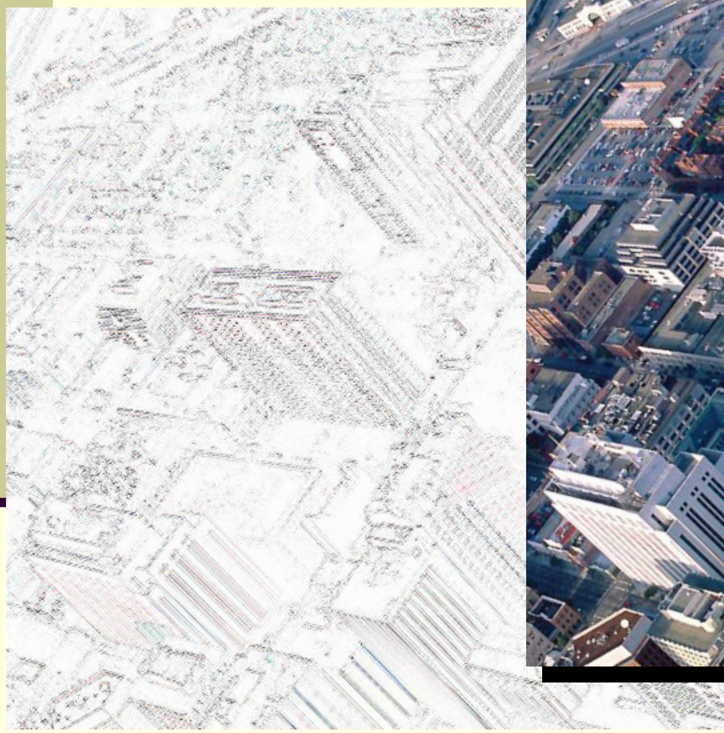


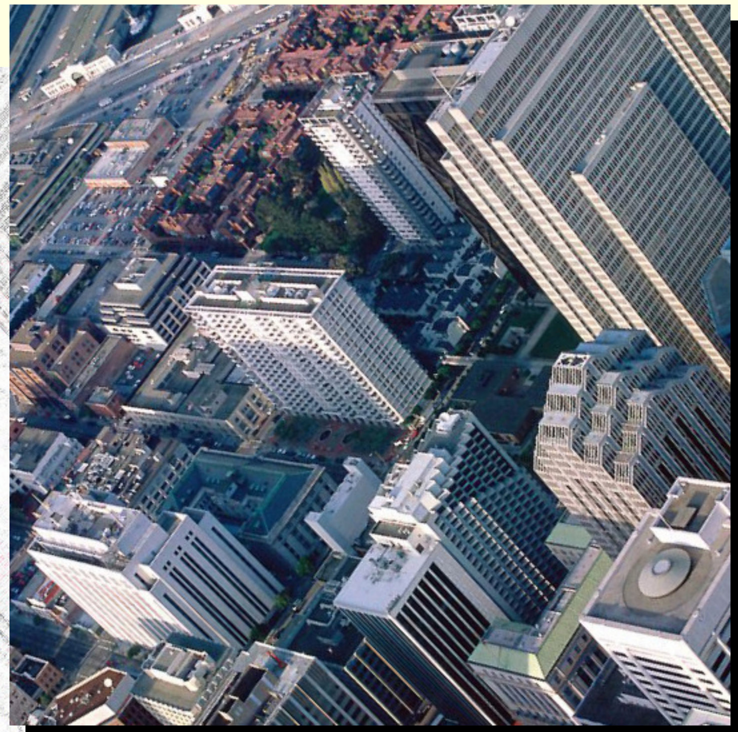
Nearest Neighbor – Bilinear

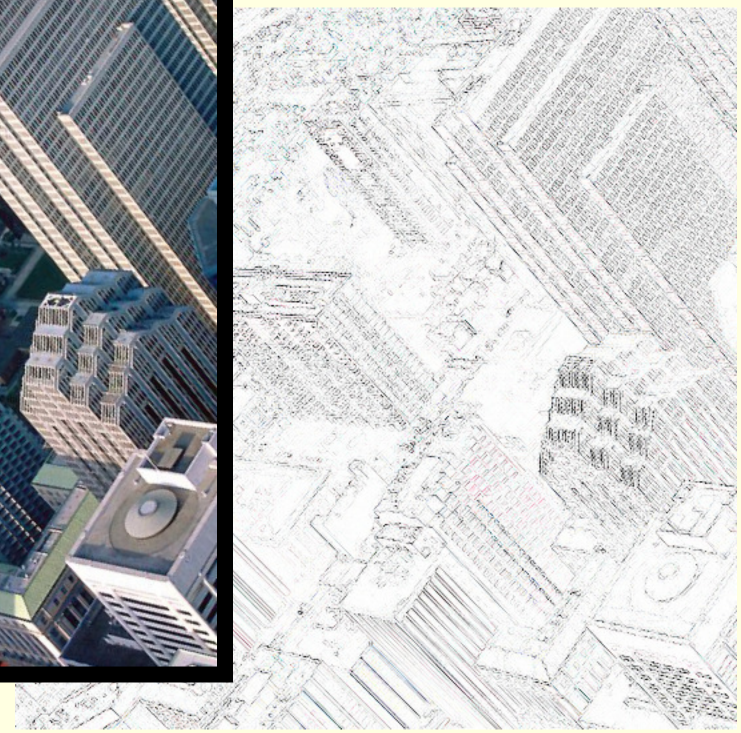
Bilinear – Bicubic

Геометрично изкривяване – пример

Ротация с бикубична интерполация





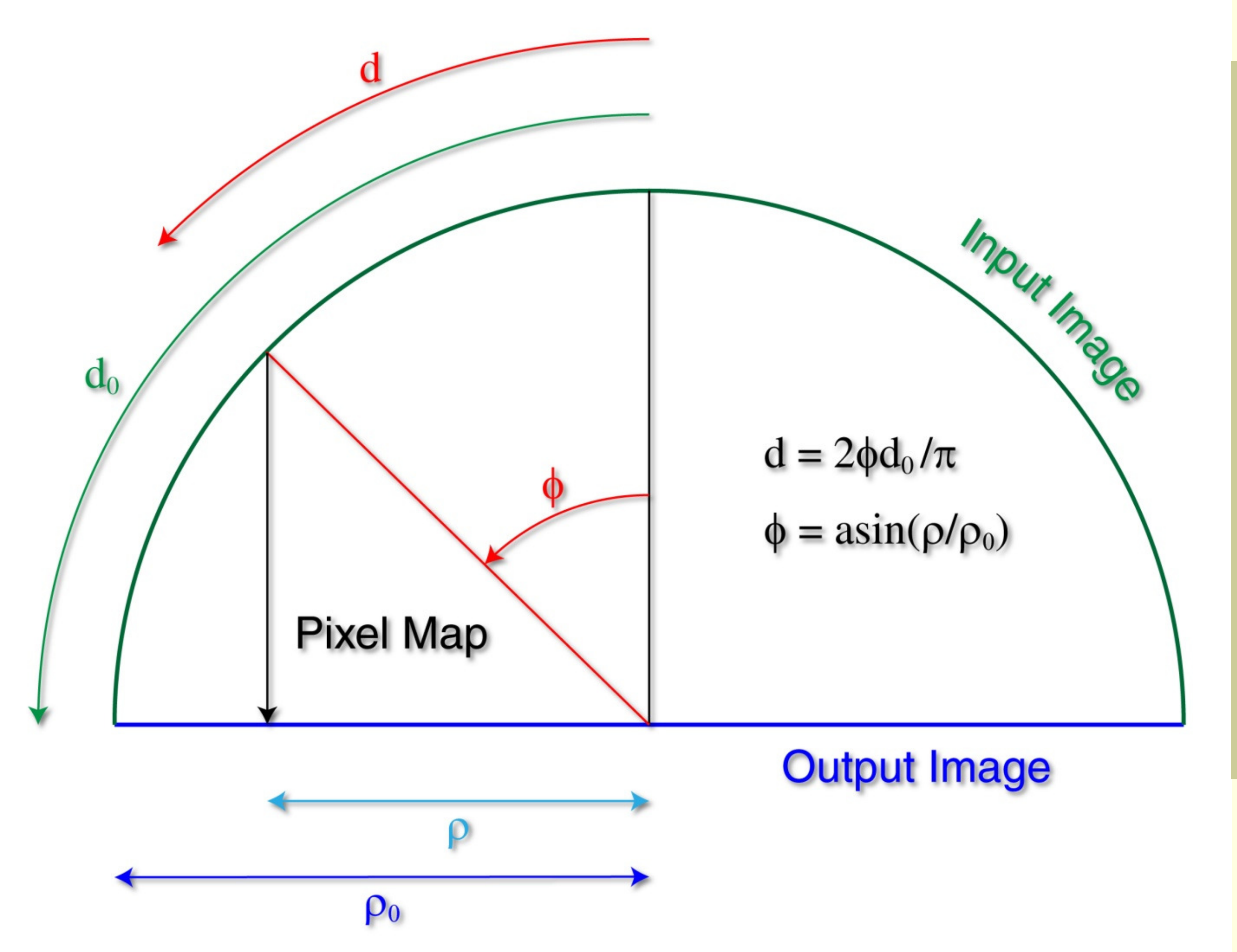


Bilinear – Bicubic

Bicubic – Nearest Neighbor

Изкривяване – пример

Изобразяване на изображение върху сфера



d: радиално разстояние от центъра на входното изображение

r: радиално разстояние от центъра на изходното изображение

Изкривяване – пример

Изобразяване на изображение върху сфера

За изходното изображение:

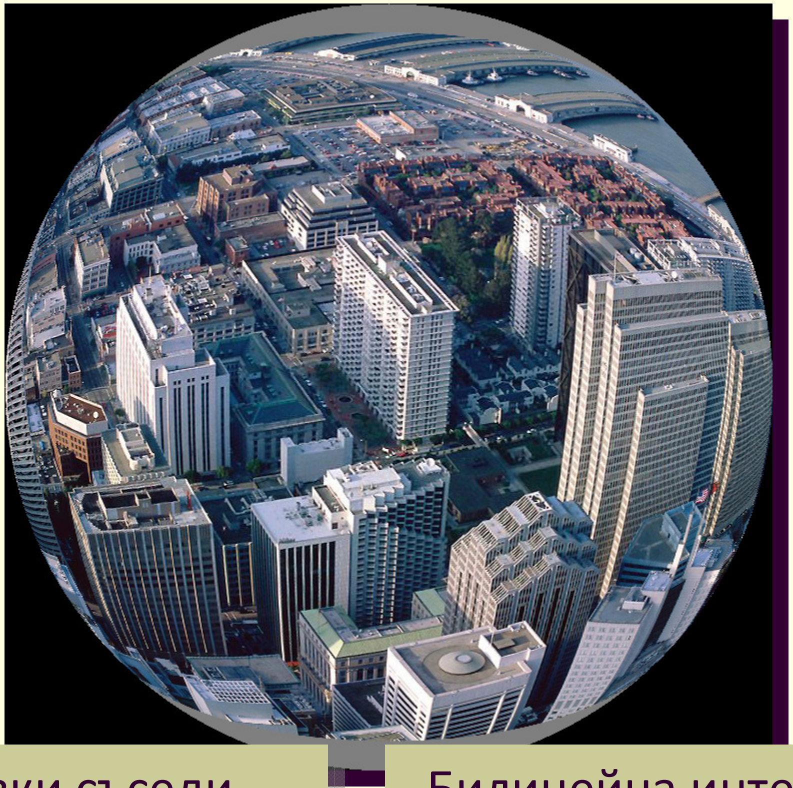
$$\rho_0 = \frac{1}{2}\min\left(R_{\text{out}}, C_{\text{out}}\right).$$

За всеки изходен пиксел:

$$\rho = \sqrt{r_{\text{out}}^2 + c_{\text{out}}^2},$$

$$\theta = \tan^{-1}\left(\tfrac{r_{\text{out}}}{c_{\text{out}}}\right),$$

$$\phi = \sin^{-1}\left(\tfrac{\rho}{\rho_0}\right).$$



За входното изображение:

$$d_0 = \frac{1}{2}\max\left(R_{\text{in}}, C_{\text{in}}\right).$$

За всеки изходен пиксел, входния пиксел е:

$$d = \tfrac{2}{\pi} d_0 \phi$$

$$r_{\text{in}} = d\sin(\theta)$$

$$c_{\text{in}} = d\cos(\theta)$$

Най-близки съседи

Билинейна интерполация

КРАЙ

Следваща тема:

Компресия на изображения

Уейвлетни преобразувания