Цифрова обработка на изображения

Геометрични операции с изображения

доц. Милена Лазарова, кат. КС, ФКСУ

Геометрични трансформации

- В цифровата обработка на изображения термина “***геометрична трансформация***” означава промяна на пространствените отношения на пикселите в изображението
  - ***rubber-sheet transformation***
    - отпечатване на изображението върху лист от гума и разтягане на листа съгласно определени правила
- Два вида операции

(1) ***геометрични операции***

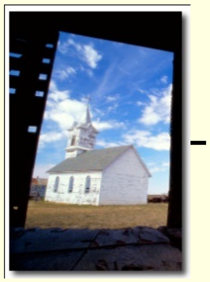
- промяна (преподреждане) на пространственото разположение на пикселите в равнината на изображението

(2) ***интензитетни интерполации***

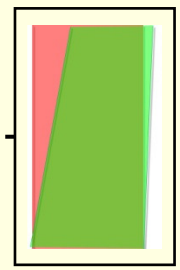
- определяне на стойности на пикселите в пространствено трансформираното изображение

Геометрични трансформации

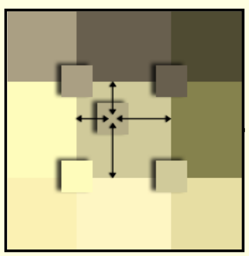
Geometric Remapping



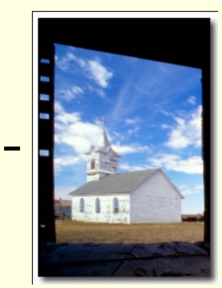
Входно изображение



Функция на изкривяване
(***warping, mapping***)



Интерполационна функция
(***resampling***)



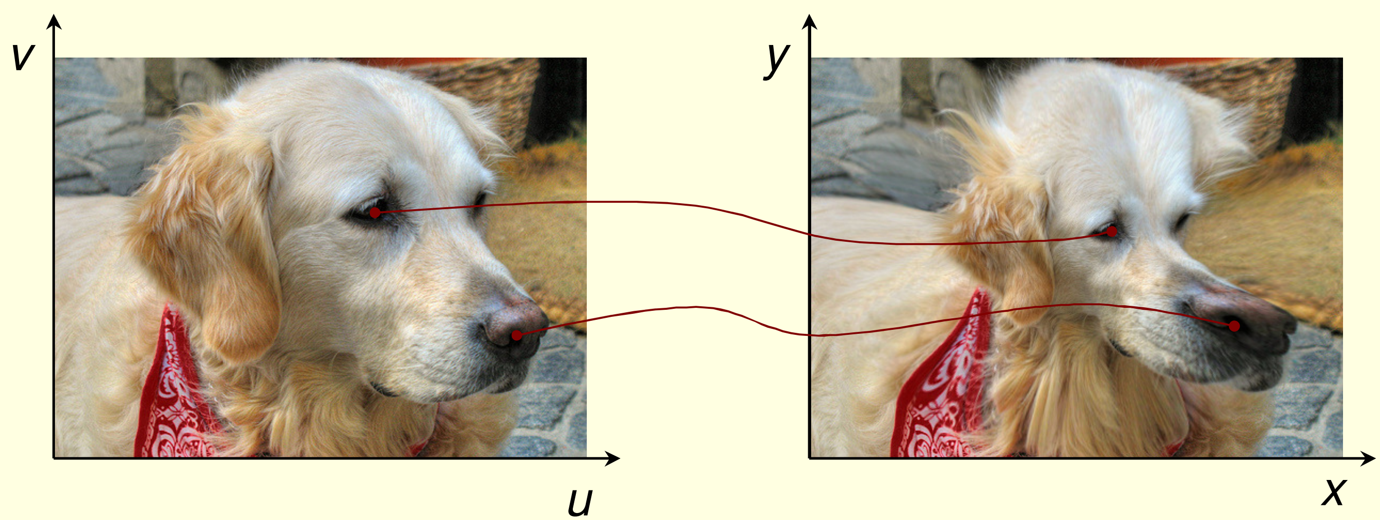
Изходно изображение

Геометрични трансформации

- Допуска се, че входното изображение I има безкрайна пространствена разделителна способност
- Изчисляват се размерите $R_{out} \times C_{out}$ на изходното изображение J
- Изчислява се функция на изкривяването Φ (warping function)
  - Създава се масив Φ с размери $R_{out} \times C_{out} \times 2$
  - За всеки пиксел в J се определя съответна точка с реални координати в (r_f, c_f) в I
  - Определя се $\Phi(r,c,1) = r_f$ and $\Phi(r,c,2) = c_f$
- Създава се интерполационна функция Θ, с която се определят стойностите на пикселите в изходното изображение на базата на стойностите в I за локална околност $\aleph(r_f, c_f)$
- Определя се $J(r,c) = \Theta\{I;\ \aleph(r_f, c_f)\}$

Геометрични операции

- ***Image Mapping***
- Определя се пространствената трансформация на входното в изходното изображение
  - дефинира се трансформацията, с която всеки изходен пиксел (x,y) се получава от входен пиксел (u,v)



Геометрични операции

- Пространствена трансформация на входното в изходното изображение
  - определя промяна на Декартовите координати от входното към изходното изображение
- Прости пространствени трансформации
  - ***транслиране***, ***мащабиране***, ***ротиране***
- Комбинация от няколко прости трансформации
  - ***произволна функция*** за преобразуване на входните координати в изходните

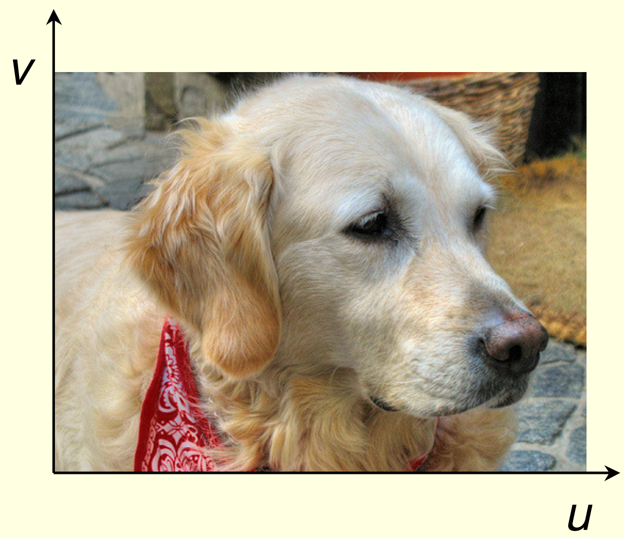
Геометрични операции

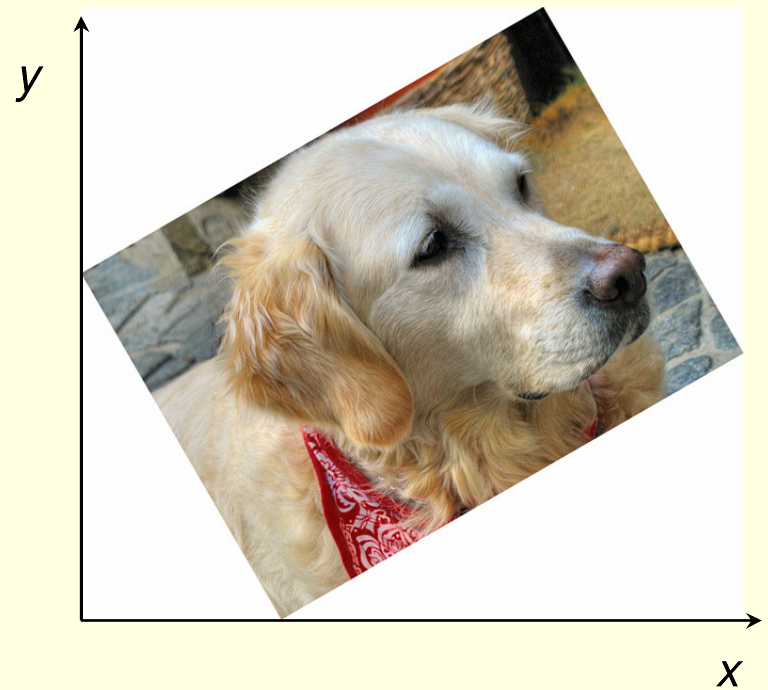
- Мащабиране
  - промяна на размера с определен коефициент μ
    - $x = \mu \cdot u$
    - $y = \mu \cdot v$

мащабиране с коефициент $\mu = 0.8$

Геометрични операции

- Ротация
  - завъртане на ъгъл θ
    - $x = u \cdot \cos\theta - v \cdot \sin\theta$
    - $y = u \cdot \sin\theta + v \cdot \cos\theta$

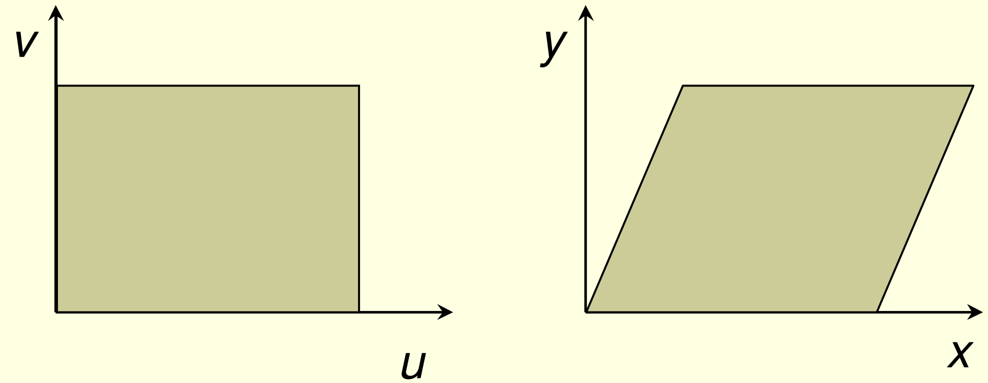




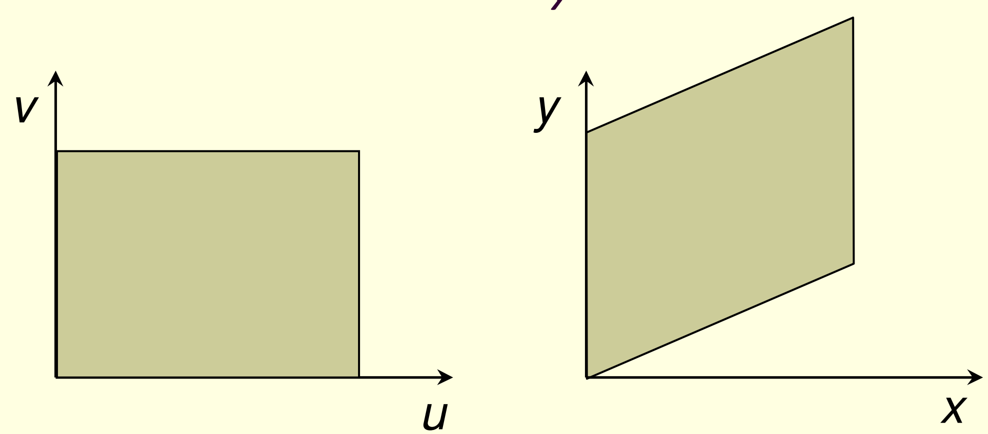
ротация на ъгъл θ = -30°

Геометрични операции

- Изкривяване по x с коефициент λ_x
  - $x = u + \lambda_x \cdot v$
  - $y = v$



- Изкривяване по y с коефициент λ_y
  - $x = u$
  - $y = v + \lambda_y \cdot u$



Геометрични операции

- Произволна функция за *u*, *v*
  - $x = f_x(u,v)$
  - $y = f_y(u,v)$

twirl

zig-zag

spherize

Геометрични операции

- Пространствената трансформация може да се изпълни със задаване на целево изходно изображение
  - ***Image Registration***

- Линейна или нелинейна функция на пространствената трансформация
  - в зависимост от реда на функцията на базата на зададени контролни точки от входното и целевото изходно изображение се определят параметрите на пространствената трансформация

Image Registration

- ***Цел***: на базата на няколко двойки съответни координати от входното и изходното изображение (контролни точки)

$$(u_{\kappa},v_{\kappa}) \rightarrow (x_{\kappa},y_{\kappa}),\ \kappa = 1, \ldots, N$$

да се определи функцията на трансформацията

$$x = f_x(u,v)$$

$$y = f_y(u,v)$$

Image Registration

- Два вида трансформации
  - ***Rigid transformation***
    - транслация
    - ротация
    - мащабиране
    - комбинация от транслация, ротация, мащабиране
  - ***Non-Rigid transformation***
    - параметрични
      - линейни, нелинейни
    - непараметрични

Image Registration

- ***Rigid transformation***
  - Ако пространствената трансформация включва транслация с коефициенти t_x и t_y, мащабиране с коефициент μ и ротация на ъгъл θ

$$x = \mu \cdot u \cdot \cos\theta - \mu \cdot v \cdot \sin\theta + t_x$$
$$y = \mu \cdot u \cdot \sin\theta + \mu \cdot v \cdot \cos\theta + t_y$$

  - то по две зададени съответни точки от входното и изходното изображение могат да се определят параметрите на трансформацията

Image Registration

- ***Non-Rigid transformation***
  - **Affine transformation**
    - линейна трансформация от първи ред
  - Пространствената трансформация се определя със следните уравнения

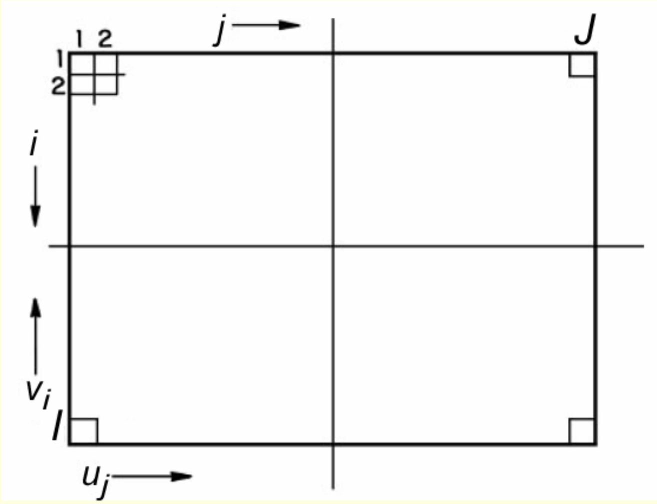
$$x = a \cdot u + b \cdot v + c$$

$$y = d \cdot u + e \cdot v + f$$

  - по три зададени съответни неколинеарни точки от входното и изходното изображение могат да се определят 6-те параметъра на трансформацията

Геометрични операции

- Входно изображение $I(i,j)$, $1 \le i \le R$, $1 \le j \le C$
  - Декартови координати за пикселите от входното изображение
    - $u_j = j - ½$
    - $v_i = C + ½ - i$
- Изходно изображение $J(p,q)$, $1 \le p \le P$, $1 \le q \le Q$
  - Декартови координати за пикселите от изходното изображение
    - $x_q = q - ½$
    - $y_p = P + ½ - p$



Геометрични операции

- Всяка геометрична операция може да се изпълни по два начина
  - ***прав (forward)***
    - за всеки пиксел от входното изображение се определя съответния пиксел от изходното изображение съгласно функцията на пространствената трансформация
      - повече от един входен пиксел може да се изобразяват в един и същи изходен пиксел
      - за определен изходен пиксел може да няма съответстващ на него входен пиксел
  - ***обратен (reverse)***
    - за всеки пиксел в изходното изображение се изчисляват координати от входното изображение съгласно функцията на пространствената трансформация
      - входните координати може да не съответстват на пиксел в изображението
        - използва се интерполация

Геометрични операции

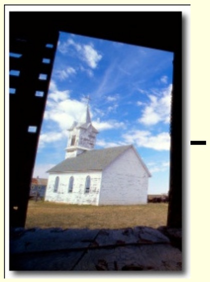
Пример:

Транслация: промяна на декартовите координати: $x_q = u_j + t_x; \; y_p = v_i + t_y$

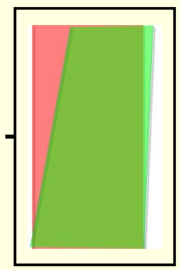
- ***прав (forward)***
  - за всеки пиксел от входното изображение (i,j) се определят (u_j,v_i), от тях се изчисляват координатите в изходното изображение (x_q,y_p), и накрая се определят (p,q)
    - $p' = i - (R - P) - t_y$
    - $q' = j + t_x$
  - p', q' може да не са цели – закръгляват се
    - може да има непопълнени пиксели в изходното изображение
- ***обратен (reverse)***
  - за всяка двойка целочислени координати в изходното изображение се изчисляват координати от входното изображение
    - $i' = \mathrm{p} - (R - P) + t_y$
    - $j' = q - t_x$
  - i', j' може да не са цели – интерполират се стойностите от входното изображение

Интерполация

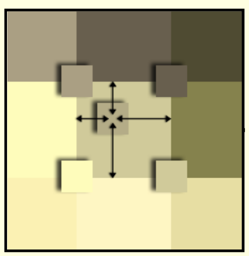
Geometric Remapping



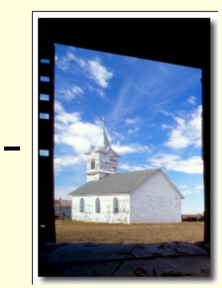
Входно изображение



Функция на изкривяване
(***warping, mapping***)



Интерполационна функция
(***resampling***)

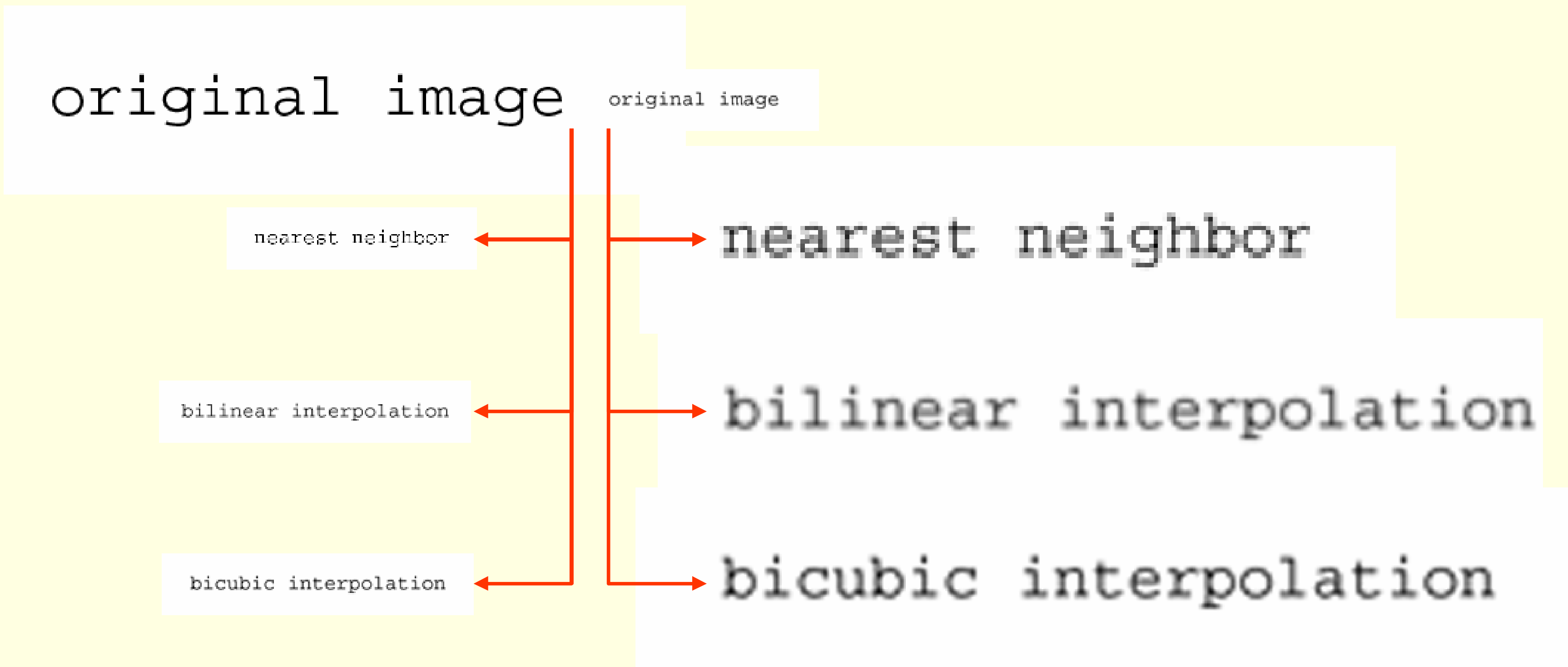


Изходно изображение

Интерполация

- Три метода за интерполация на стойностите на пикселите в изображение
  - ***Най-близък съсед***
    - Nearest neighbor
  - ***Билинейна интерполация***
    - Bilinear interpolation
  - ***Бикубична интерполация***
    - Bicubic interpolation

Интерполяция



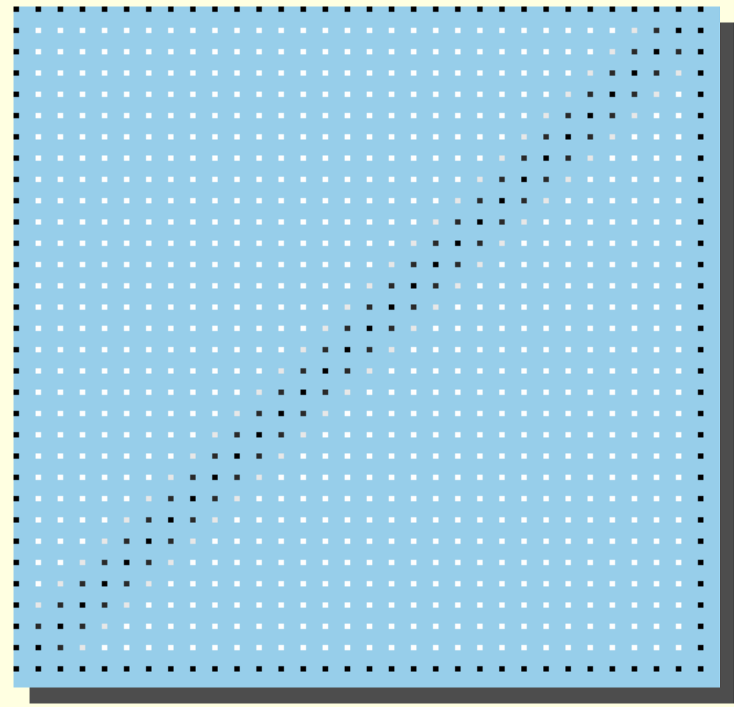
Повторение/премахване

- Най-простият подход
  - използва се при промяна на размерите на изображение (мащабиране)
- Увеличаване размера на изображение чрез повторение на пиксели
  - ***pixel replication***
- Намаляване размера на изображение чрез премахване на пиксели
  - ***pixel decimation***

Повторение/премахване

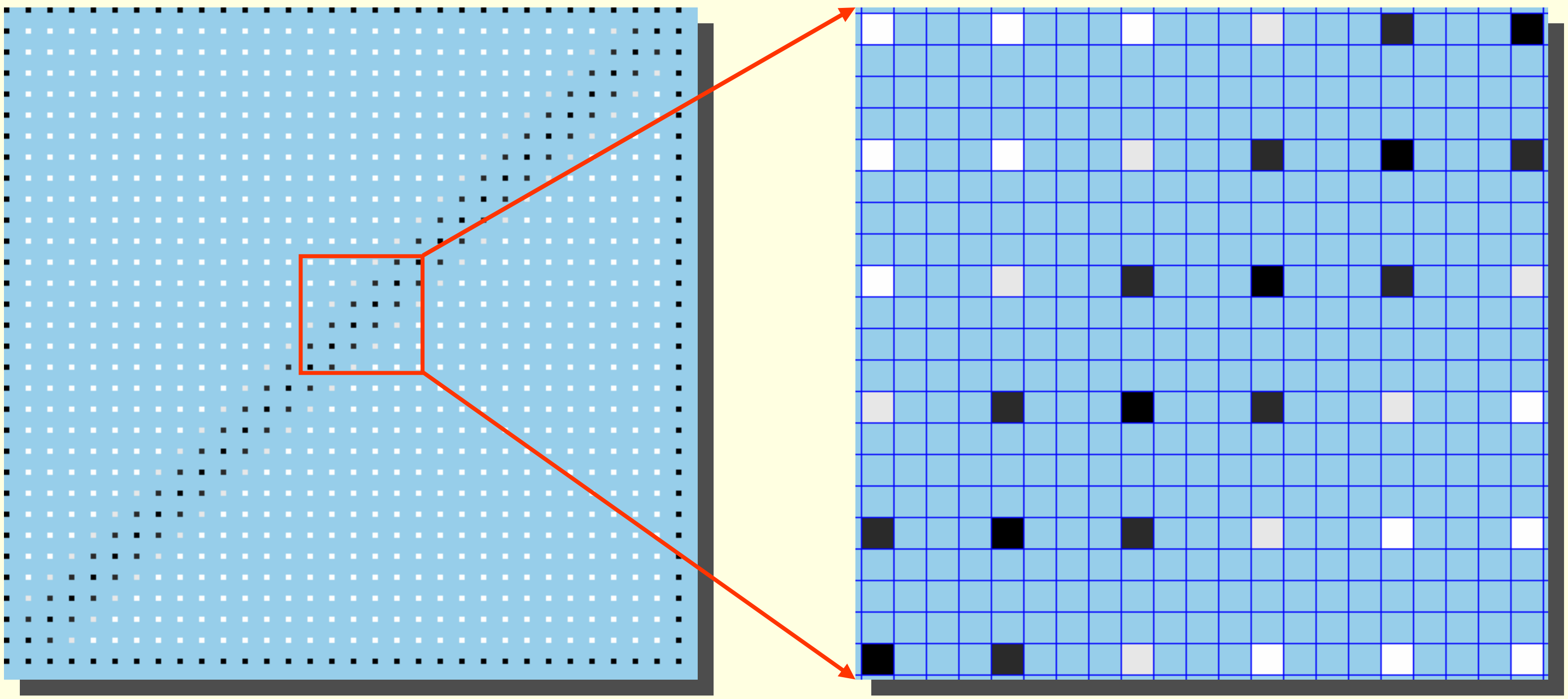
Увеличаване

Изходното изображение е с 4 пъти по-големи размери от оригиналното



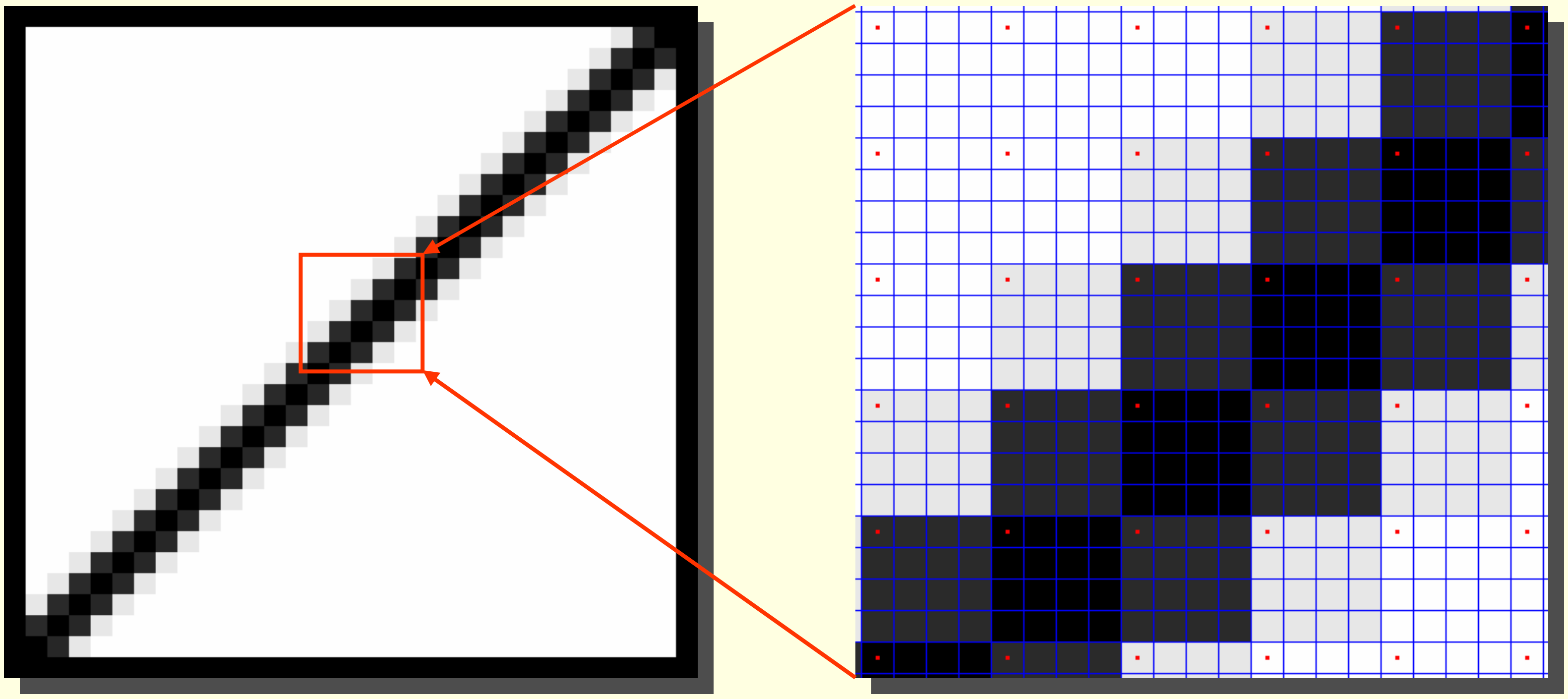
Всеки 4ти пиксел на всеки 4ти ред има стойността на пиксел от оригиналното изображение

Повторение/премахване

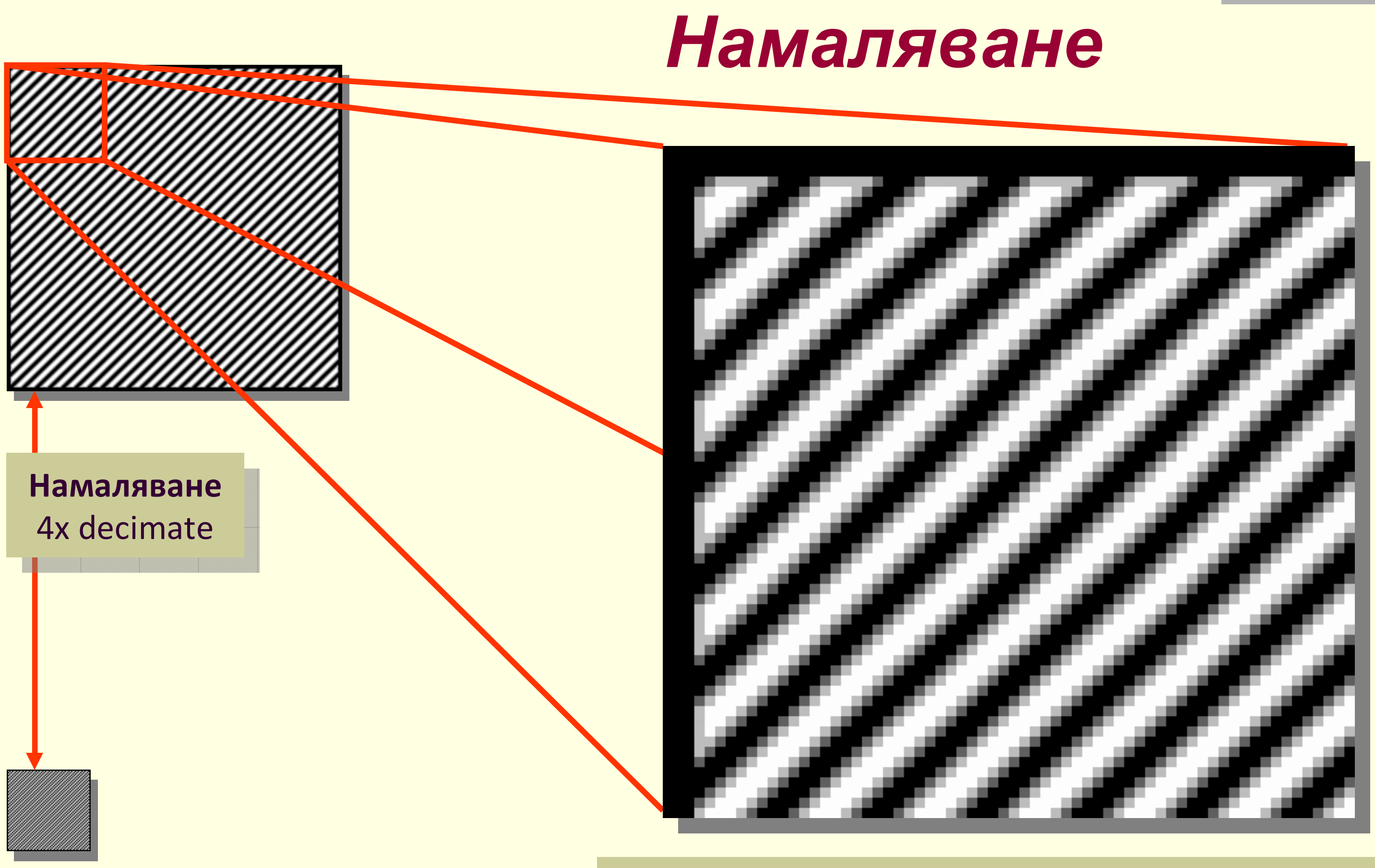


Повторение/премахване

Стойността на всеки пиксел от оригиналното изображение се повтаря 15 пъти: създава се нов по-голям “пиксел”



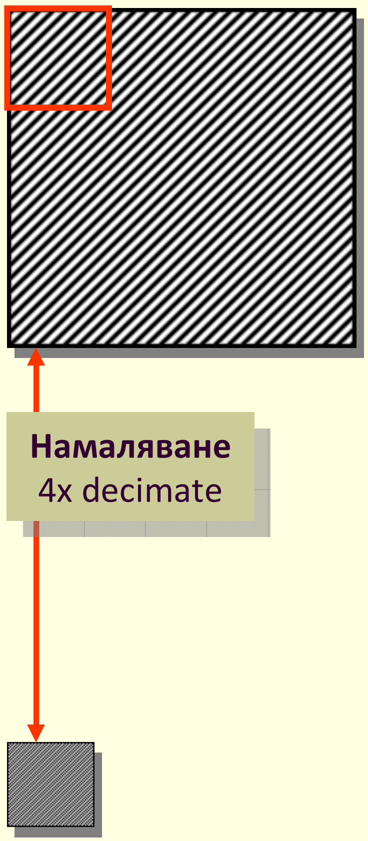
Повторение/премахване

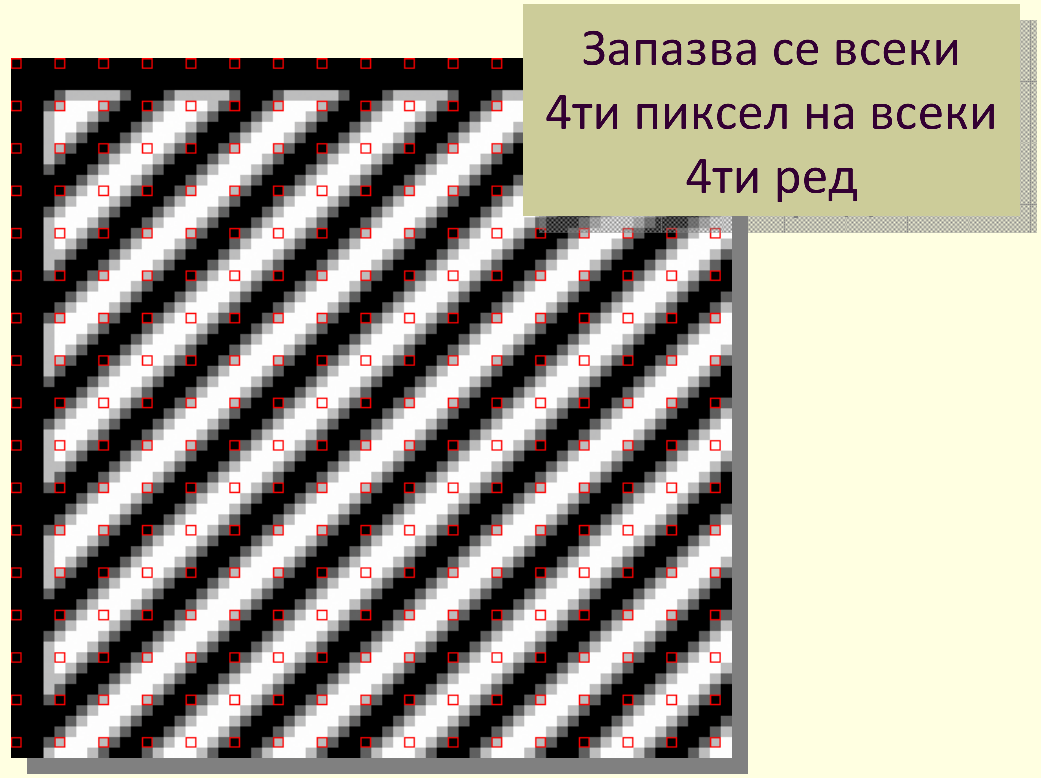


Намаляване с коефициент *n*:
всеки n-ти пиксел на всеки n-ти ред

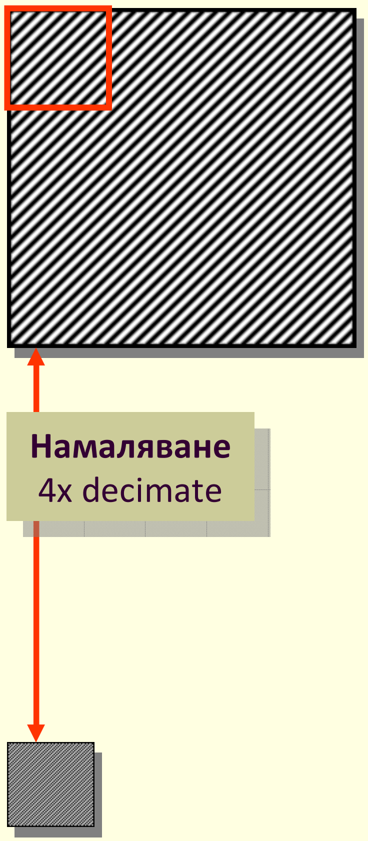
Повторение/премахване

Намаляване



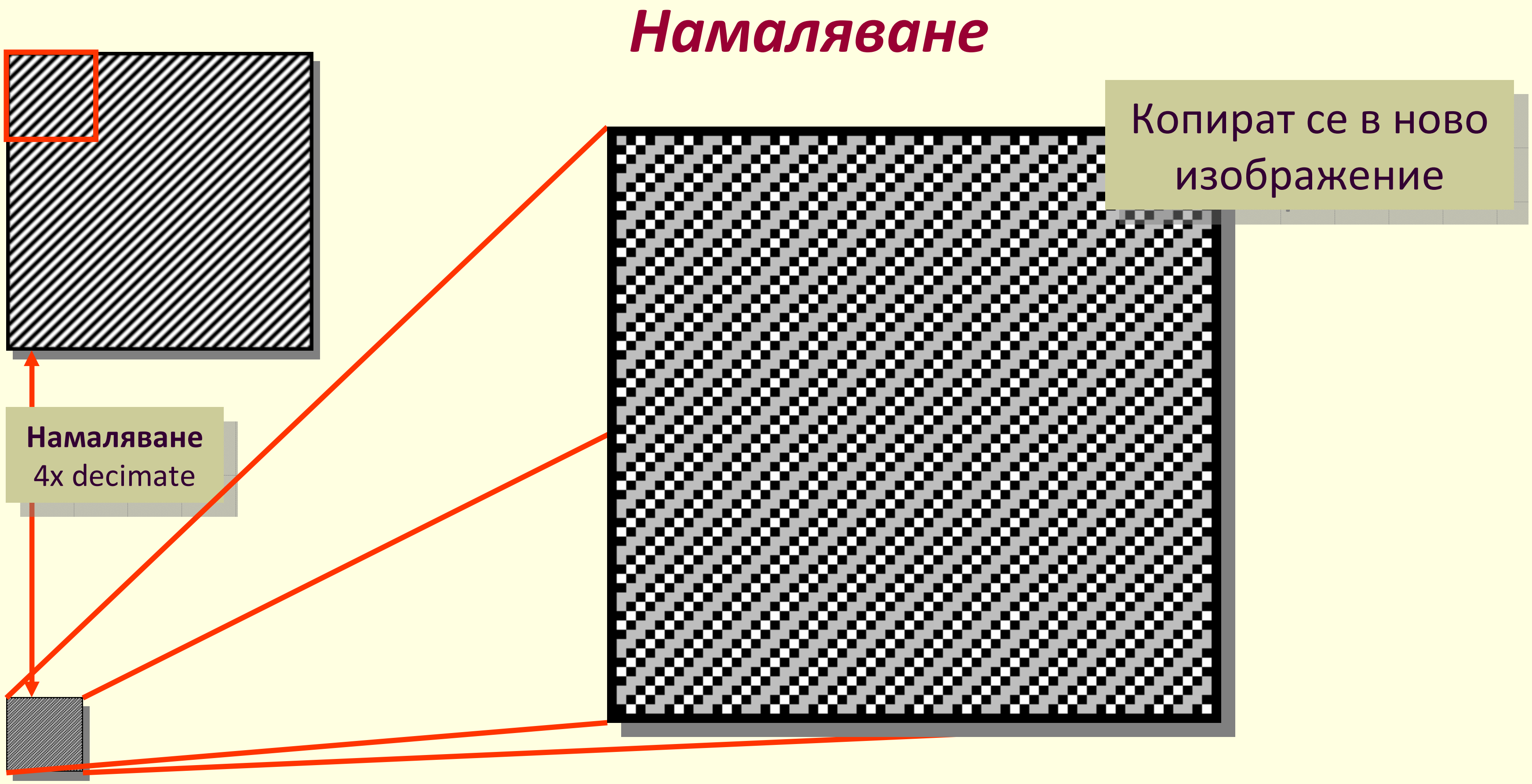


Повторение/премахване



Намаляване

Повторение/премахване

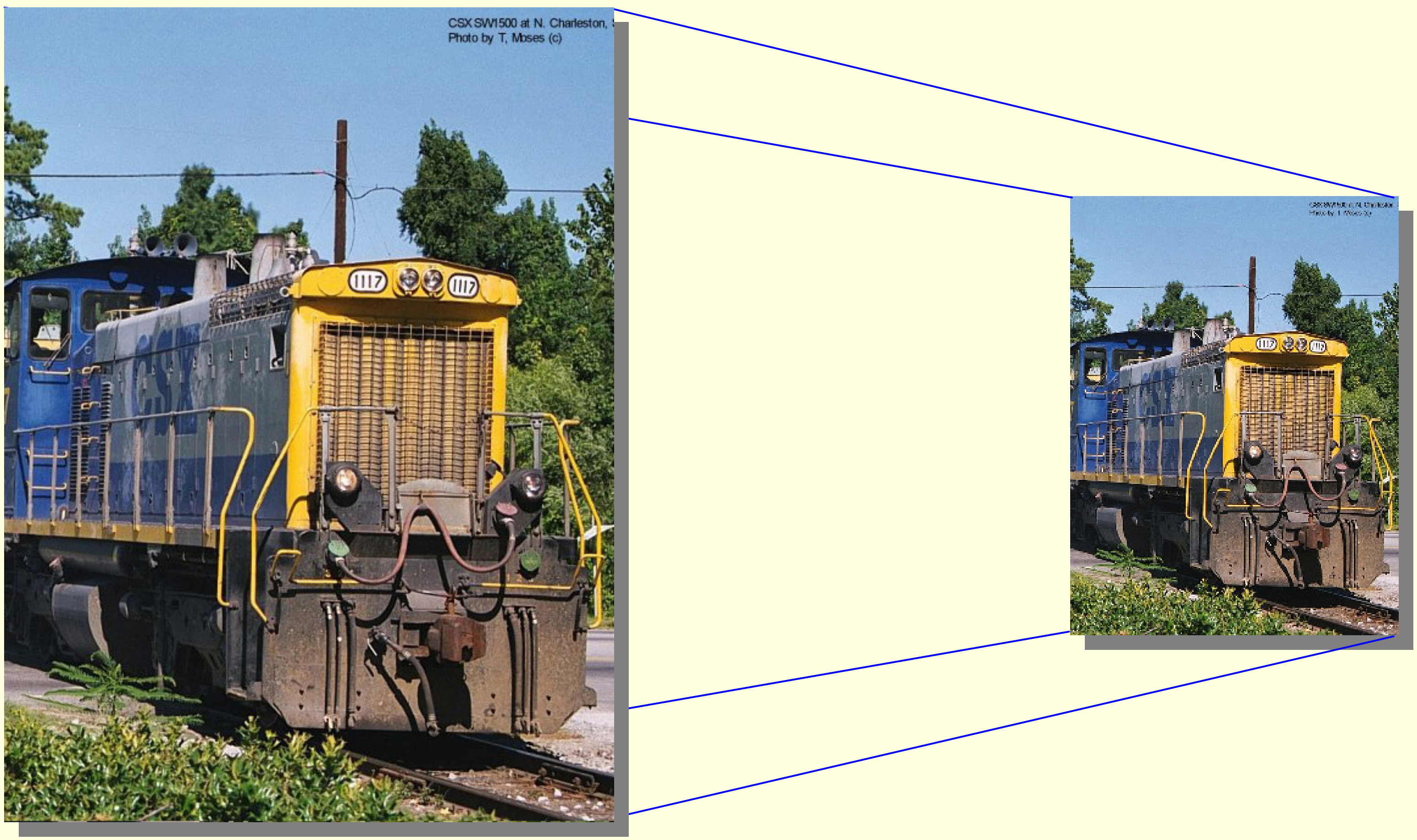


Повторение/премахване

- Може да доведе до “нащърбване”

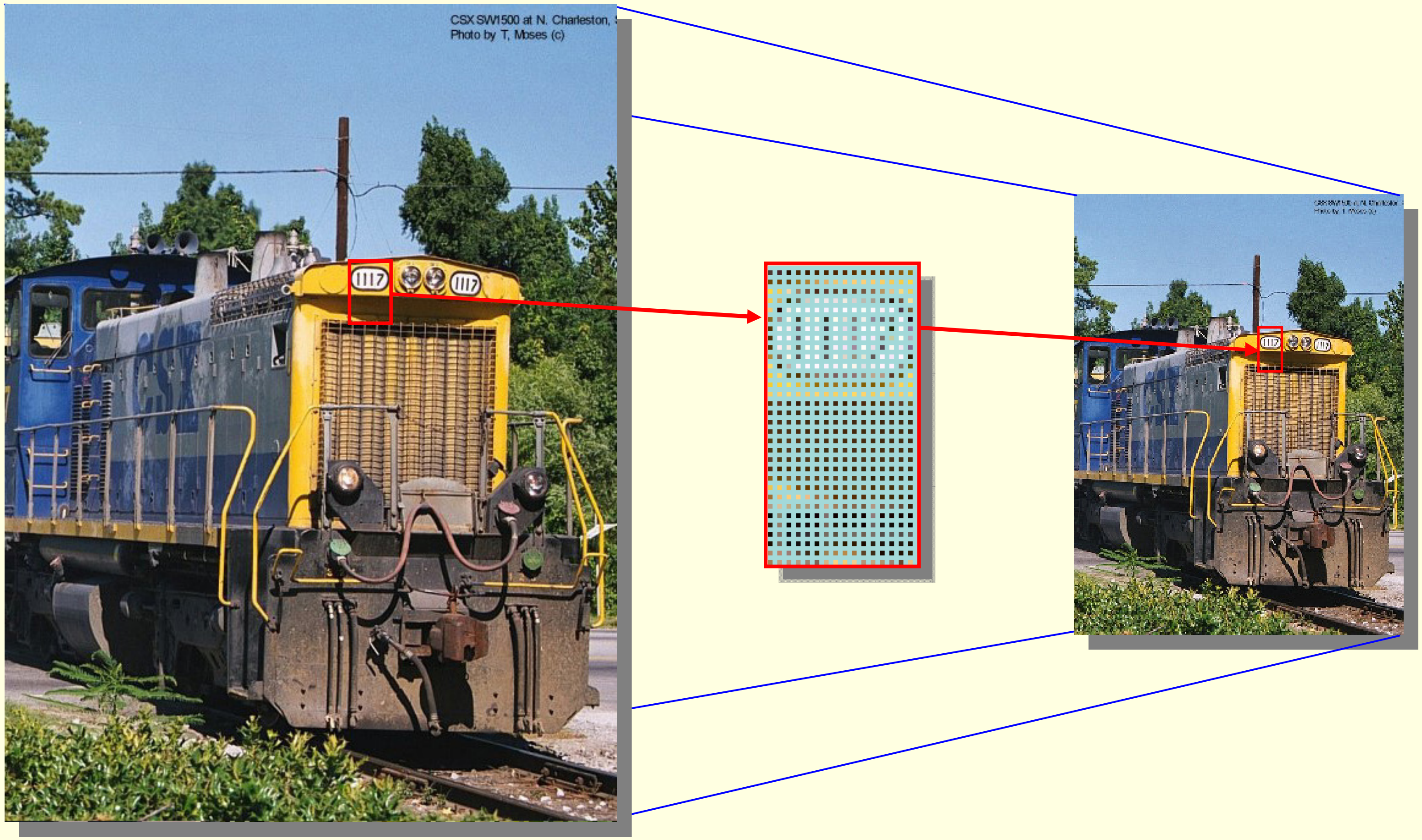
Повторение/премахване

- ***Downsampling (decimation)***
  - изходното изображение се състои от всеки четен пиксел от всеки четен ред



Повторение/премахване

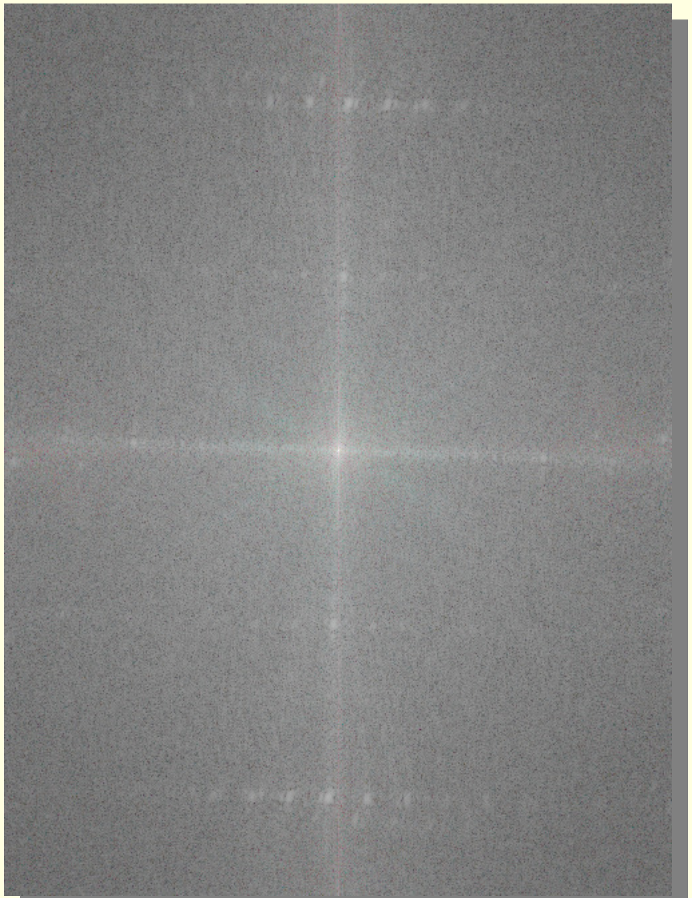
- ***Downsampling (decimation)***
  - изходното изображение се състои от всеки четен пиксел от всеки четен ред



Повторение/премахване

- Спектрална функция в Дискретно преобразувание на Фурие

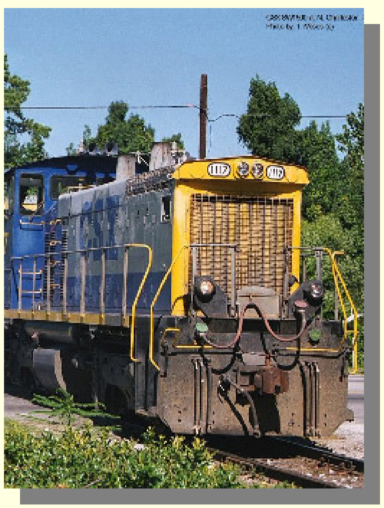
DFT



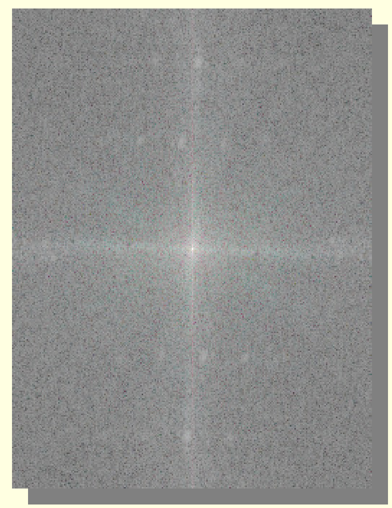
Повторение/премахване

- Спектралната функция в дискретното преобразуване на Фурие има същия размер като изображението, за която е получена

намалено изображение



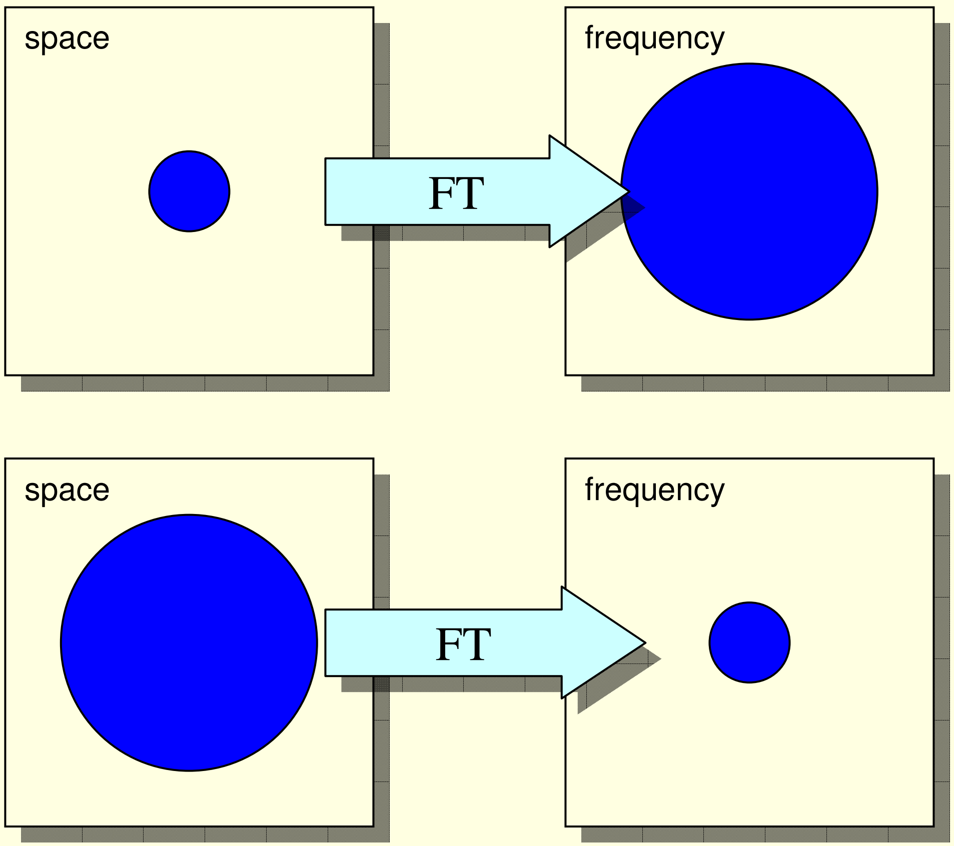
DFT



спектрална функция

Uncertainty Relation

- Малък обект в пространството на изображението има голям обхват в честотната област и обратно



- Ако $F\{I\}(v,u)$ е Фурие трансформацията на изображението $I(r,c)$
- то изображението $J(r/a,c/b)$ има Фурие трансформация

$$|ab|\ F\{I\}(av,bu)$$

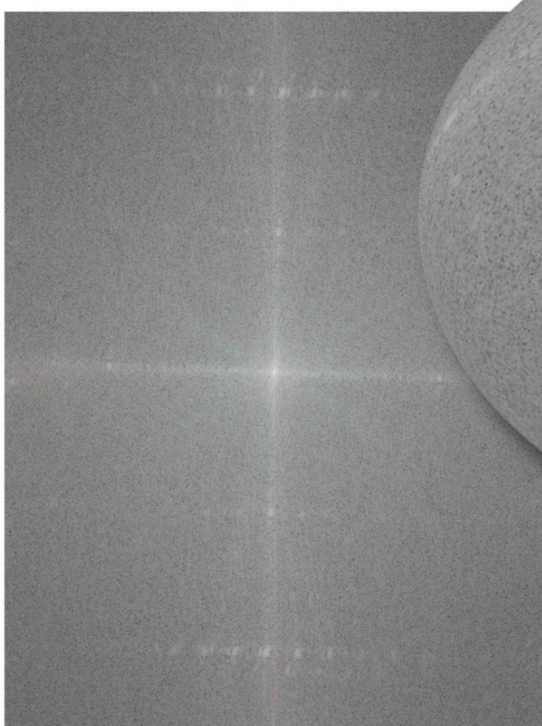
Повторение/премахване

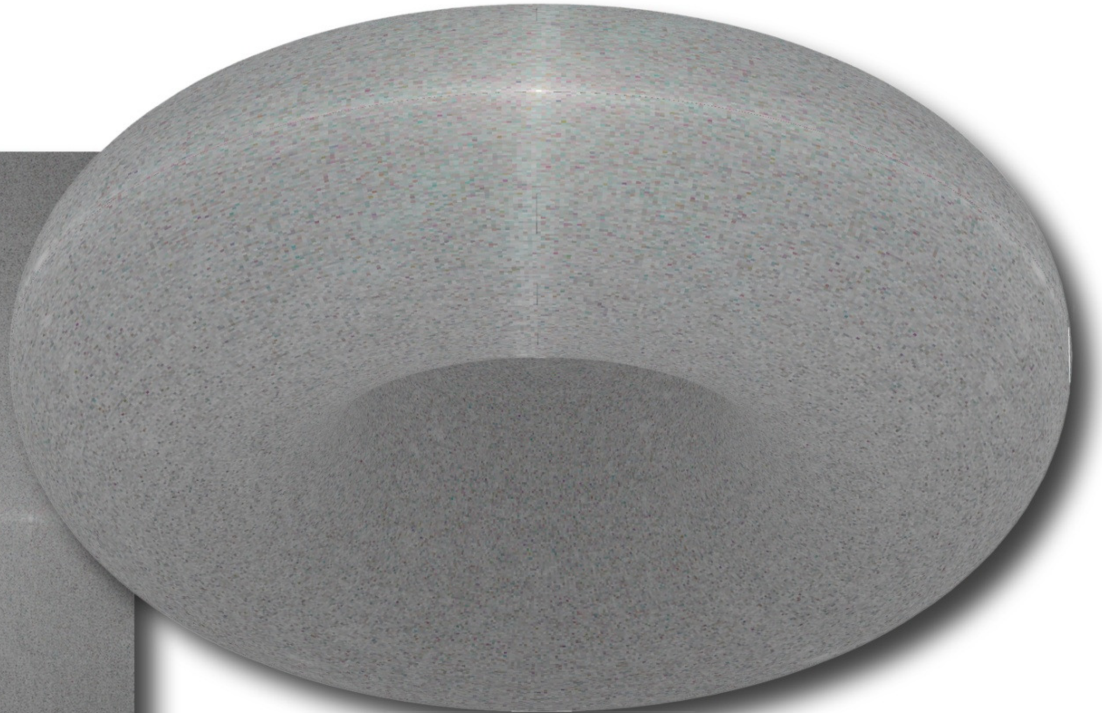
- Ефект върху Фурие преобразуванието на изображение при мащабиране
  - (1) Мащабирането (за намаляване размера) на изображението I с размери $R \times C$ с коефициент n води до получаването на изображение J с размери $\lfloor R/n \rfloor \times \lfloor C/n \rfloor$
  - (2) Дискретното преобразуване на Фурие за изображение J има същите размери като тези на изображението
  - (3) Съгласно “uncertainty relation” Фурие преобразуванието на J има размери $n \lfloor R/n \rfloor \times n \lfloor C/n \rfloor = R \times C$
- Как едновременно са изпълнени тези три факта?
- Фурие преобразуванието на J се препокрива в честотната област
  - тъй като Фурие трансформацията на дискретно изображение се дефинира за тороид
- ***Aliasing***

Aliasing

Дискретното Фурие преобразуване се получава при допускането, че изображението има тороидална топология

Aliasing



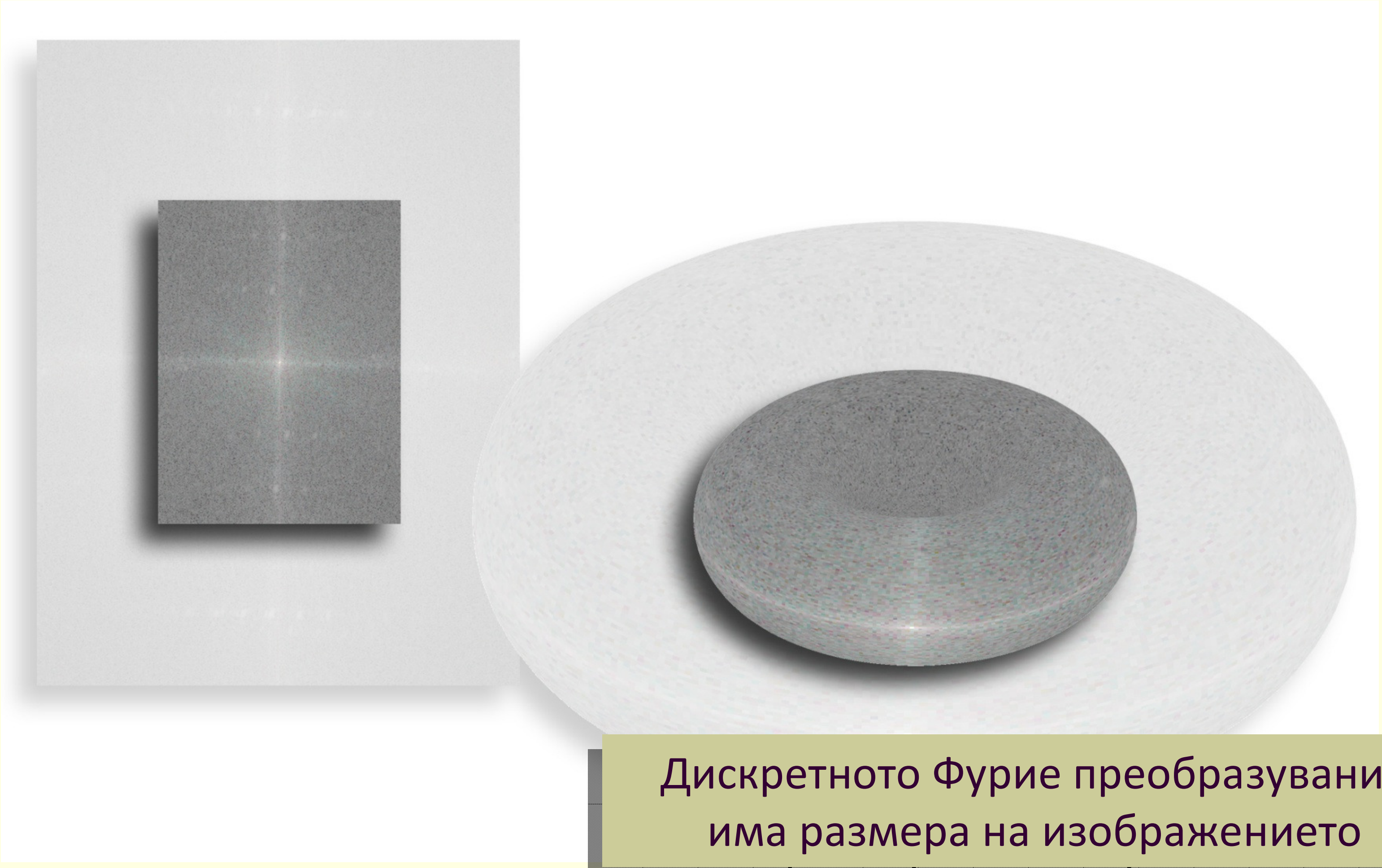


Дискретното Фурие преобразувание
също има тороидална топология

Aliasing

Намаляването на размерите на изображението означава намаляване на размера на тороида

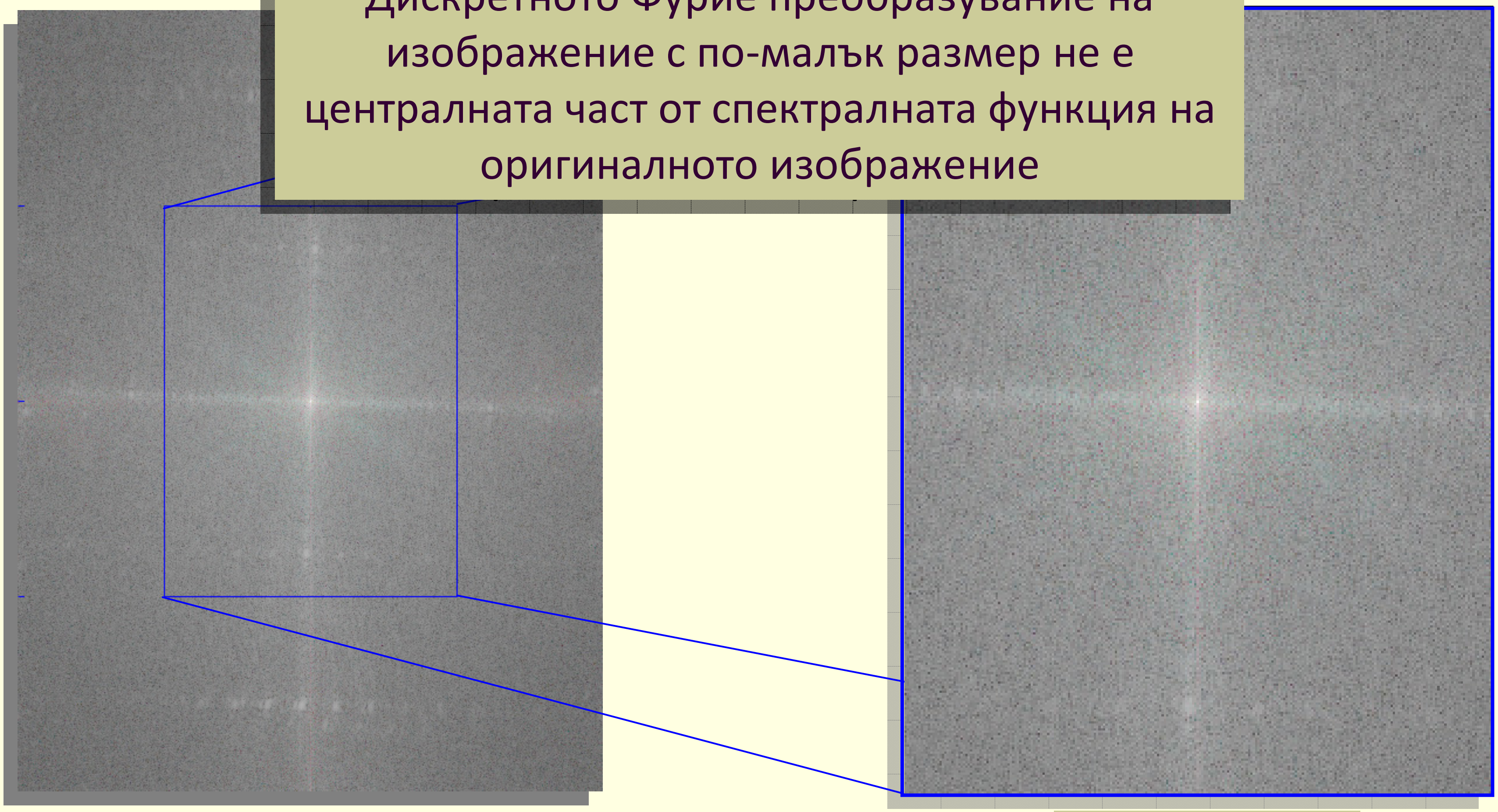
Aliasing



Дискретното Фурие преобразуване има размера на изображението

Aliasing

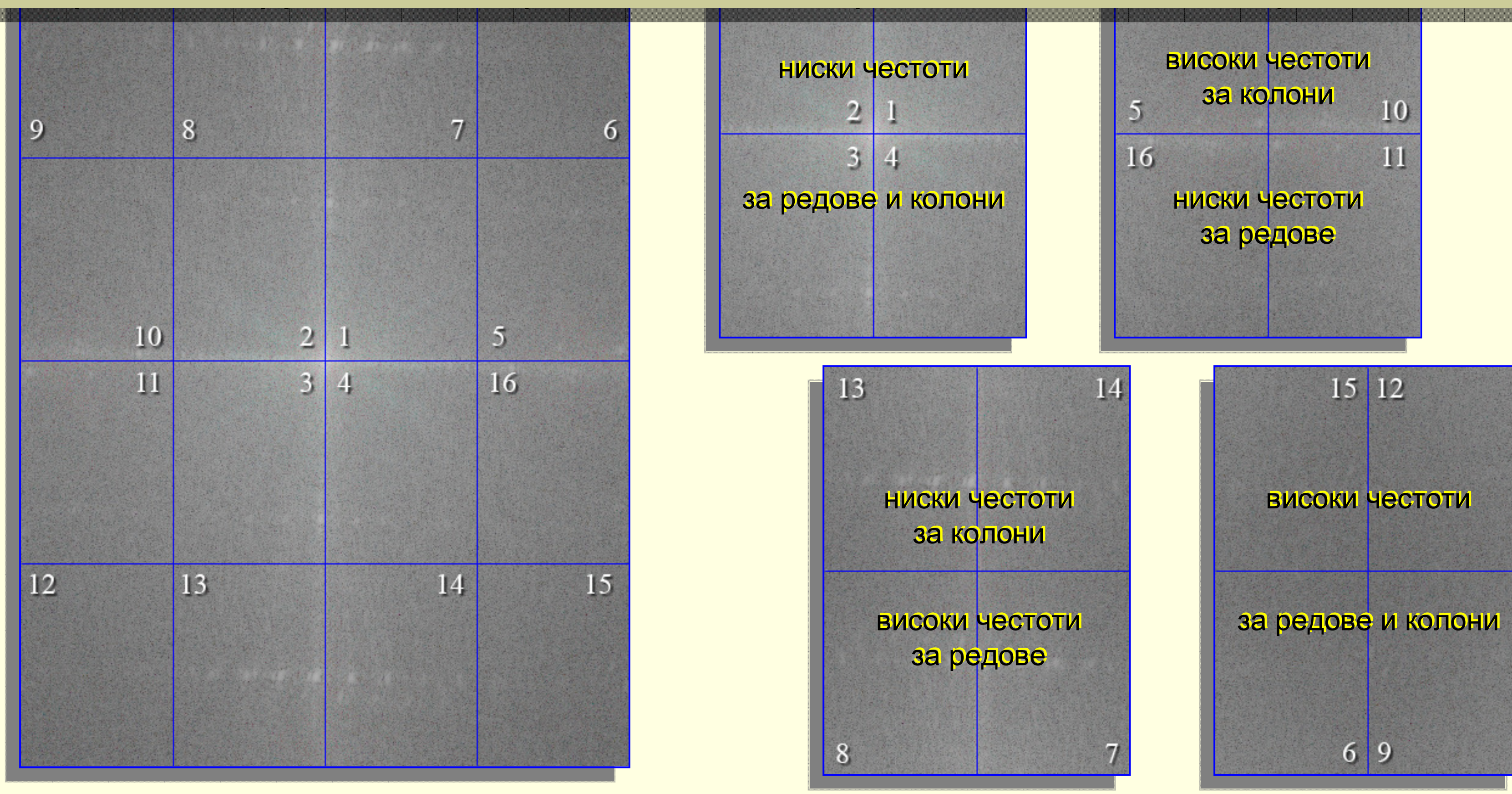
Дискретното Фурие преобразувание на изображение с по-малък размер не е централната част от спектралната функция на оригиналното изображение



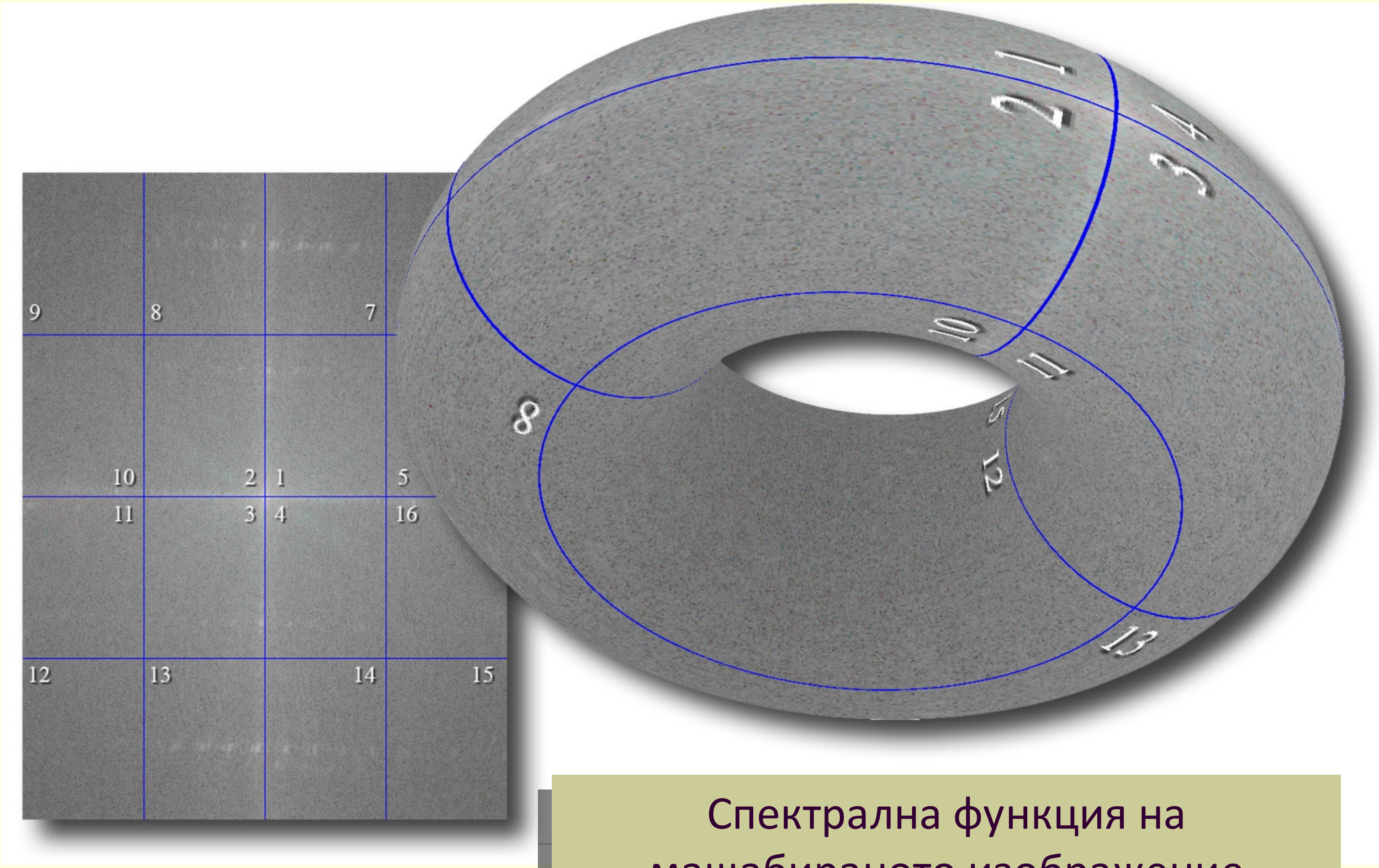
zoom 2x

Aliasing

За да се определи спектралната функция на мащабираното изображение, спектралната функция на оригиналното се разделя на няколко региона

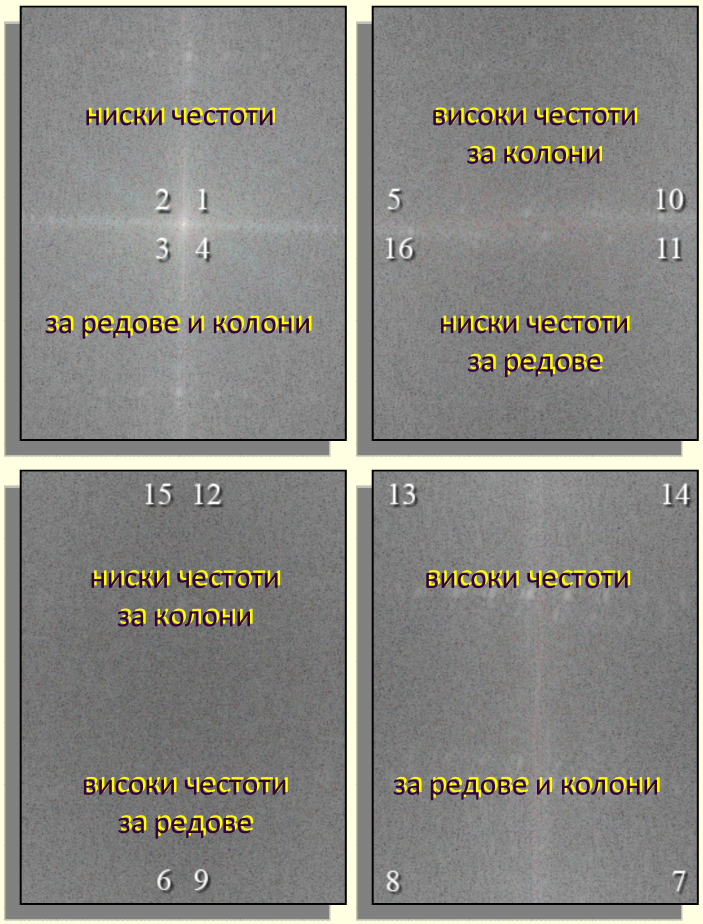


Aliasing

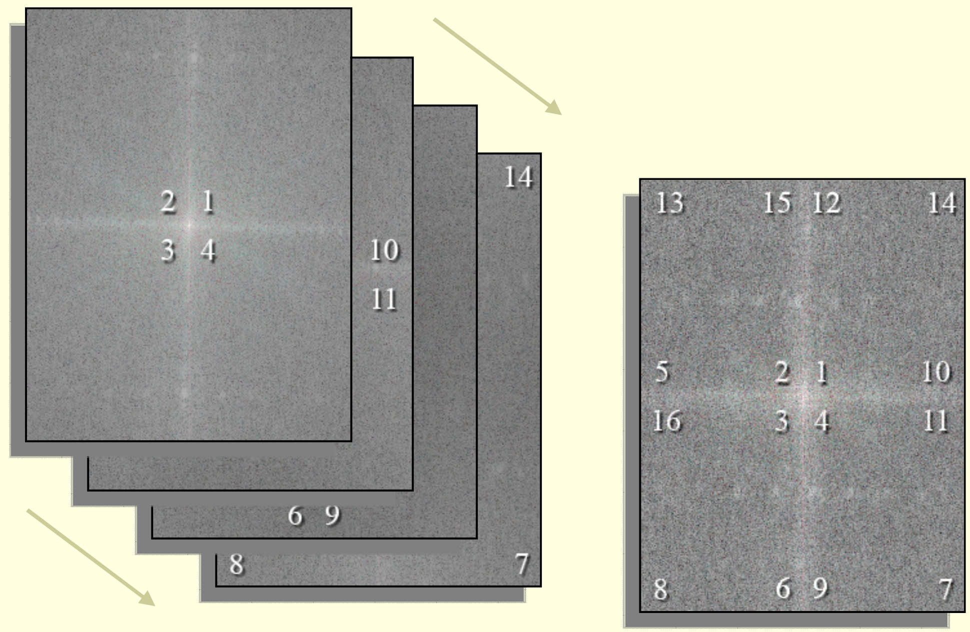


Спектрална функция на
мащабираното изображение

Aliasing



Четири отделни части на спектрална функция на мащабираното изображение, всяка от които формира тороид



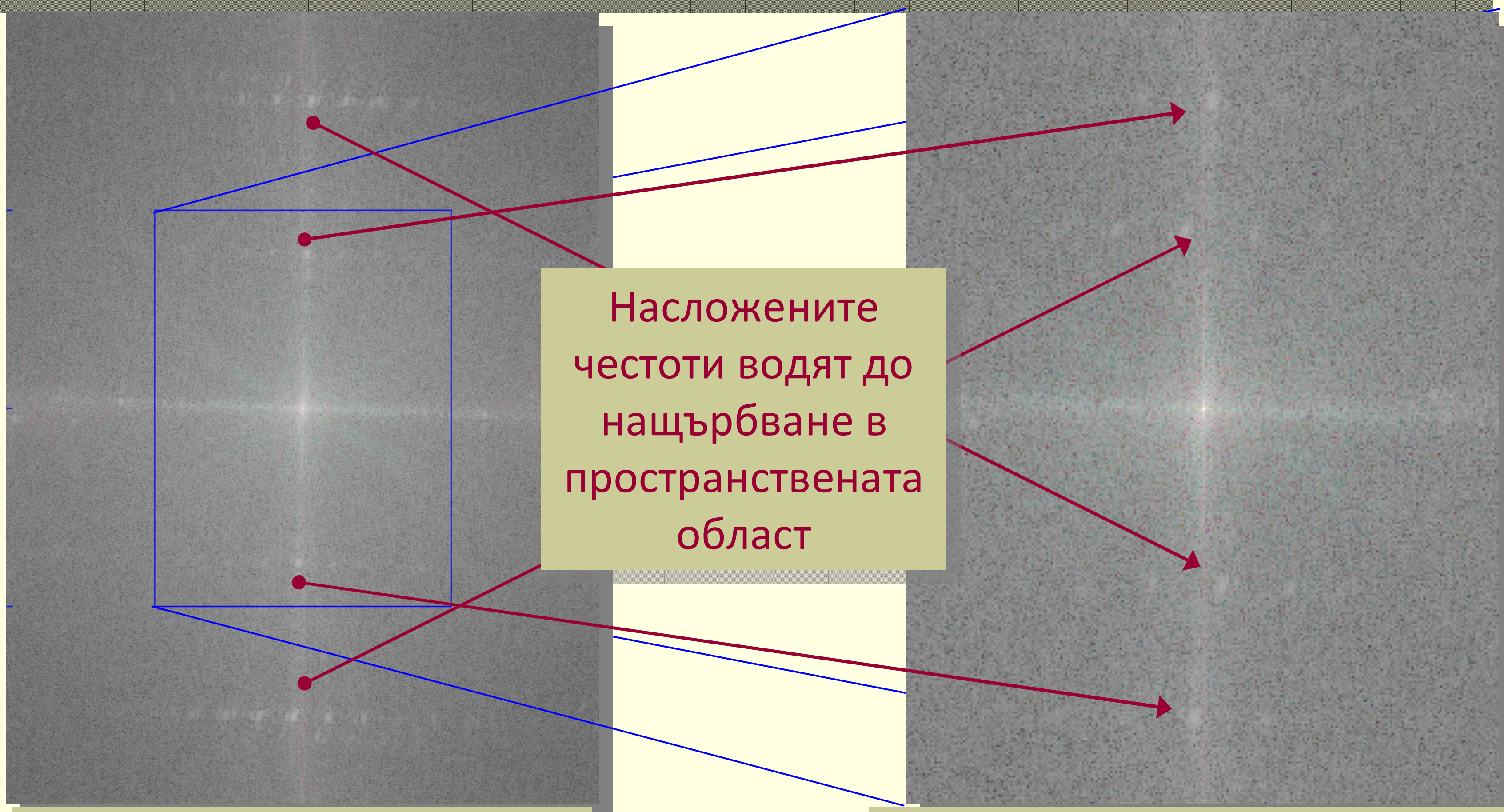
Четирите тороида се наслагват в един

Aliasing

Четирите наложени тороида наслагват високите честоти върху ниските честоти

Aliasing

Спектрална функция на оригинално и мащабирано изображение



оригинално изображение

намалено два пъти (zoom 2x)

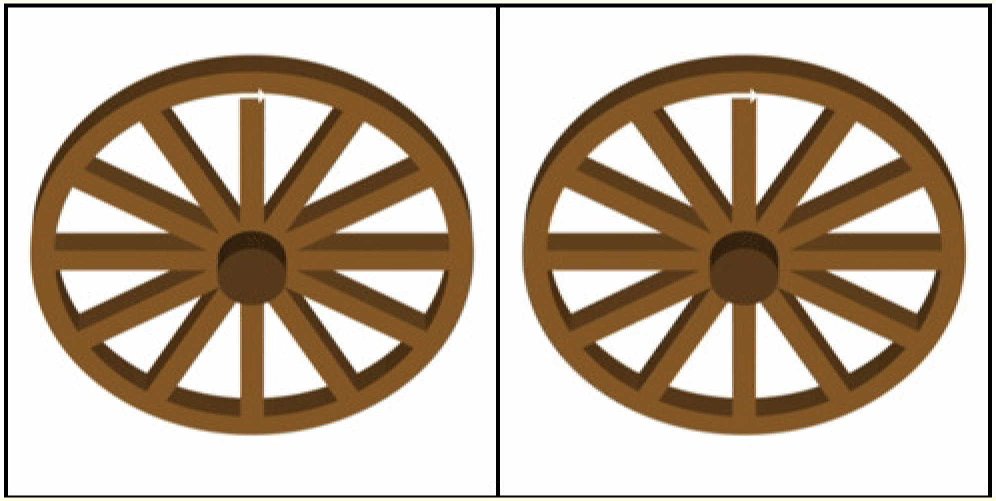
Aliasing

оригинално изображение

намалено два пъти (zoom 2x)

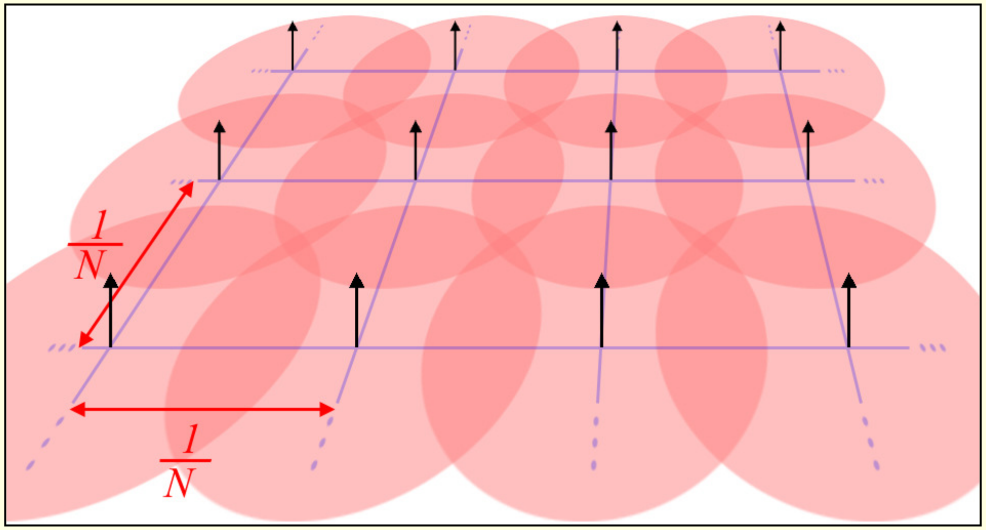
Aliasing

- Ефект “***wagon wheel***”
  - колело се върти по часовниковата стрелка
  - илюзия за въртене в обратна посока



Aliasing

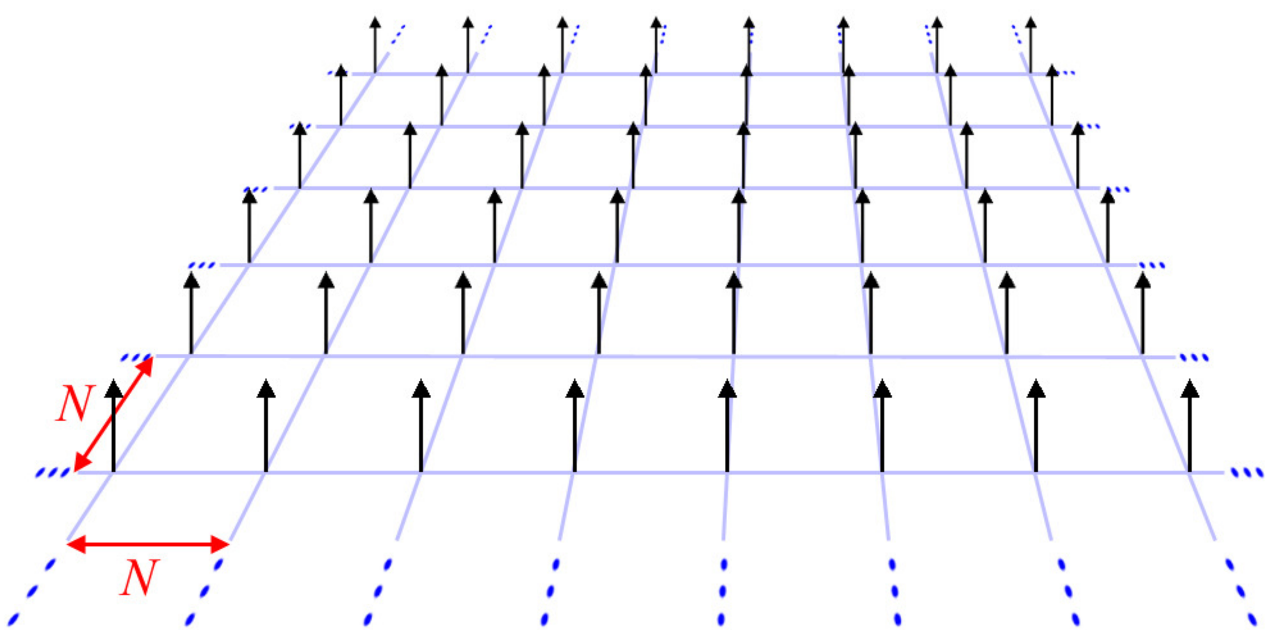
- Обяснението за aliasing се базира на пространственото дискретизиране (sampling) и конволюционните свойства на Фурие трансформацията



Aliasing

- Дискретизиращата функция (sampling function) се състои от множество импулси равномерно разположени в регулярна решетка

$$\mathrm{samp}_N(r,c) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(r - jN)\,\delta(c - kN)$$

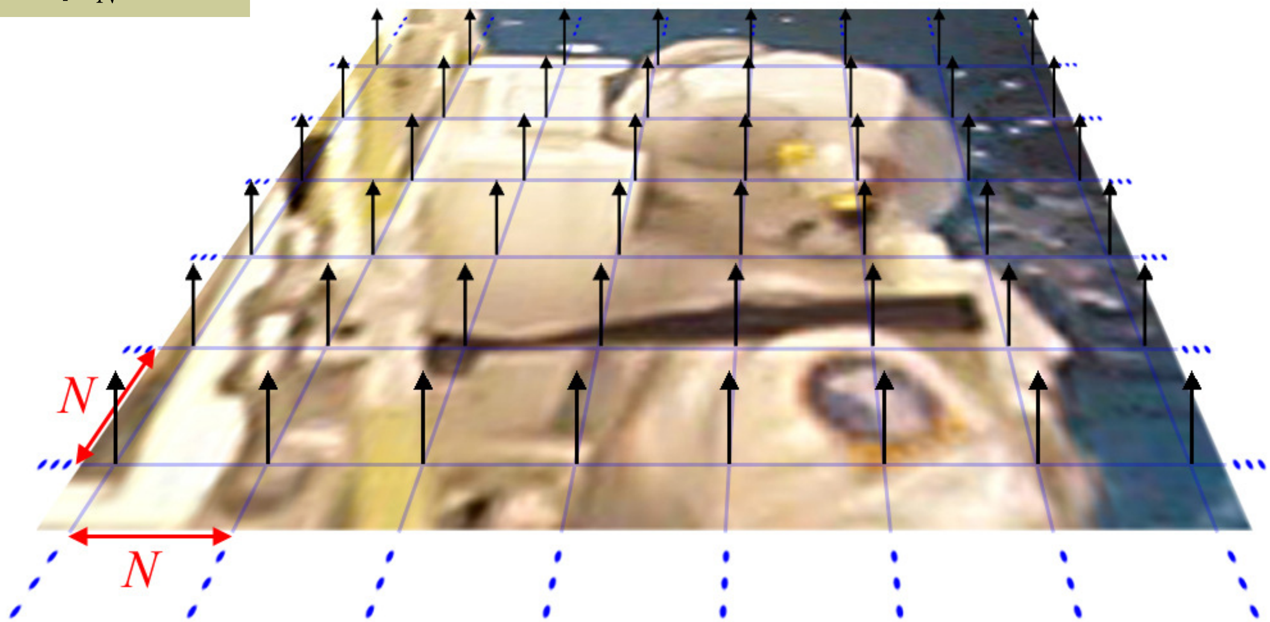


Aliasing

- Пространственото дискретизиране на изображението се свежда до умножаване на стойностите му с дискретизиращата функция

$$\mathrm{samp}_N(r,c) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta(r - jN)\,\delta(c - kN)$$

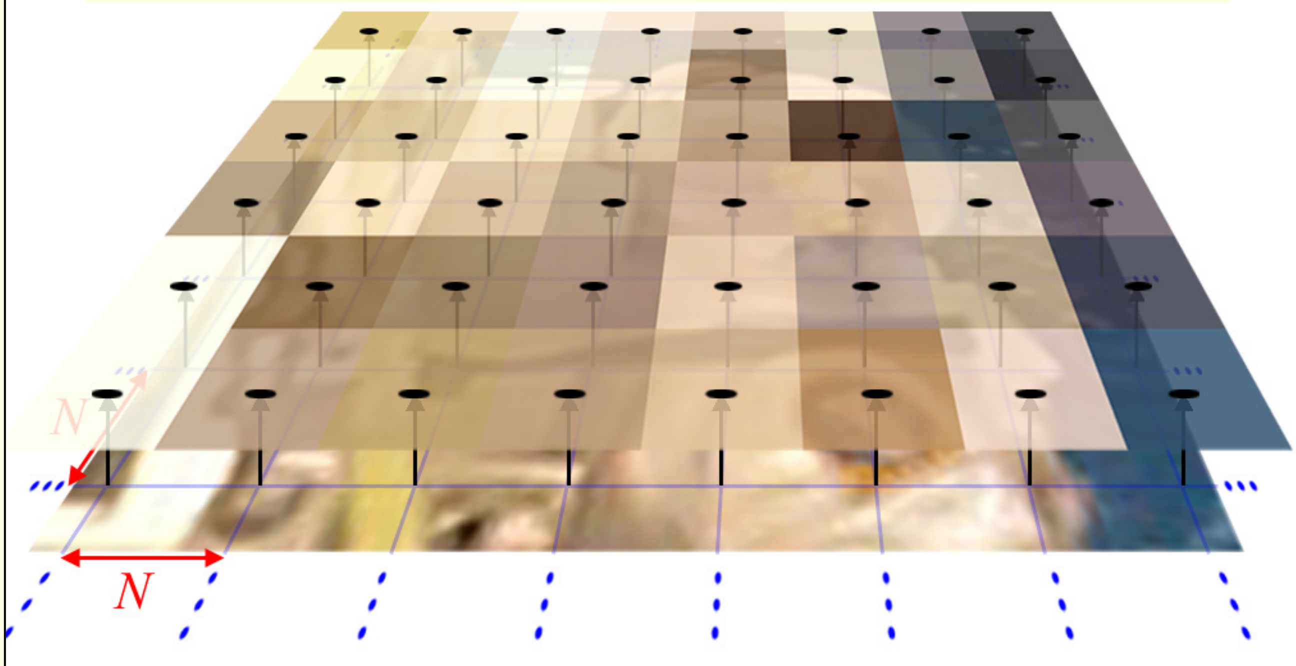
$$I(r,c) \cdot \mathrm{samp}_N(r,c)$$



Aliasing

- В пространствената област пикселите имат стойностите на оригиналното изображение и местоположението на импулсите

$$\text{samp}_N\{I\}(r,c) = I(r,c)\cdot\sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\delta(r-jN)\cdot\delta(c-kN)$$



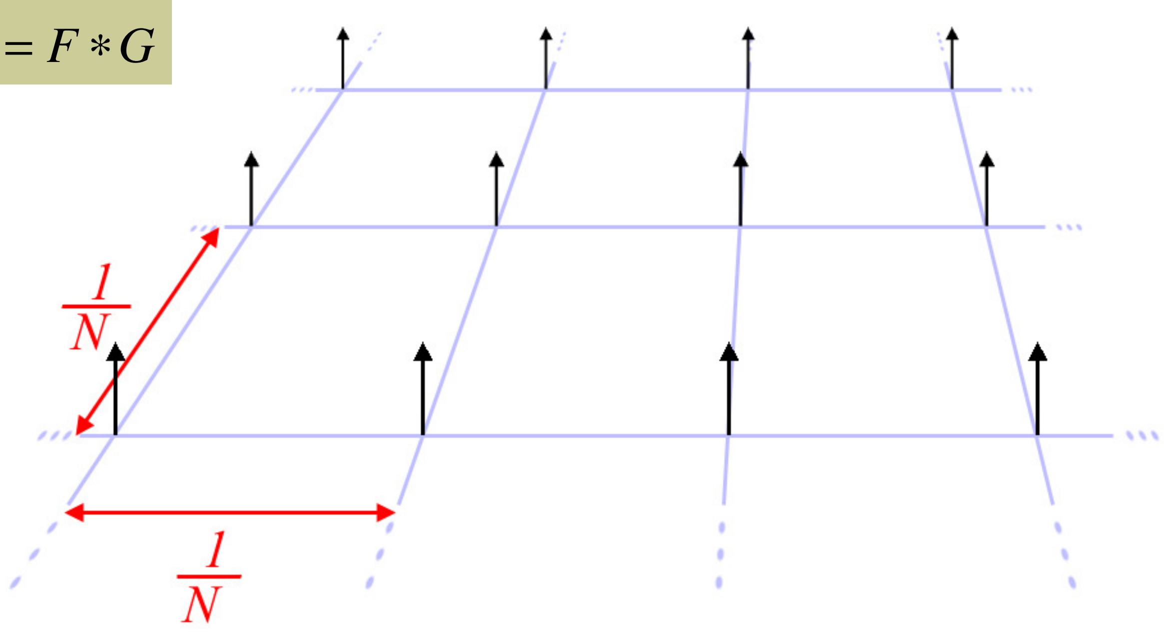
Aliasing

- Фурие трансформацията на дискретизиращата функция е същата функция с амплитуда на имулсите 1/N
  - поради конволюционните свойства на Фурие трансформацията

$$\mathbf{F}\{f * g\} = F \cdot G.$$

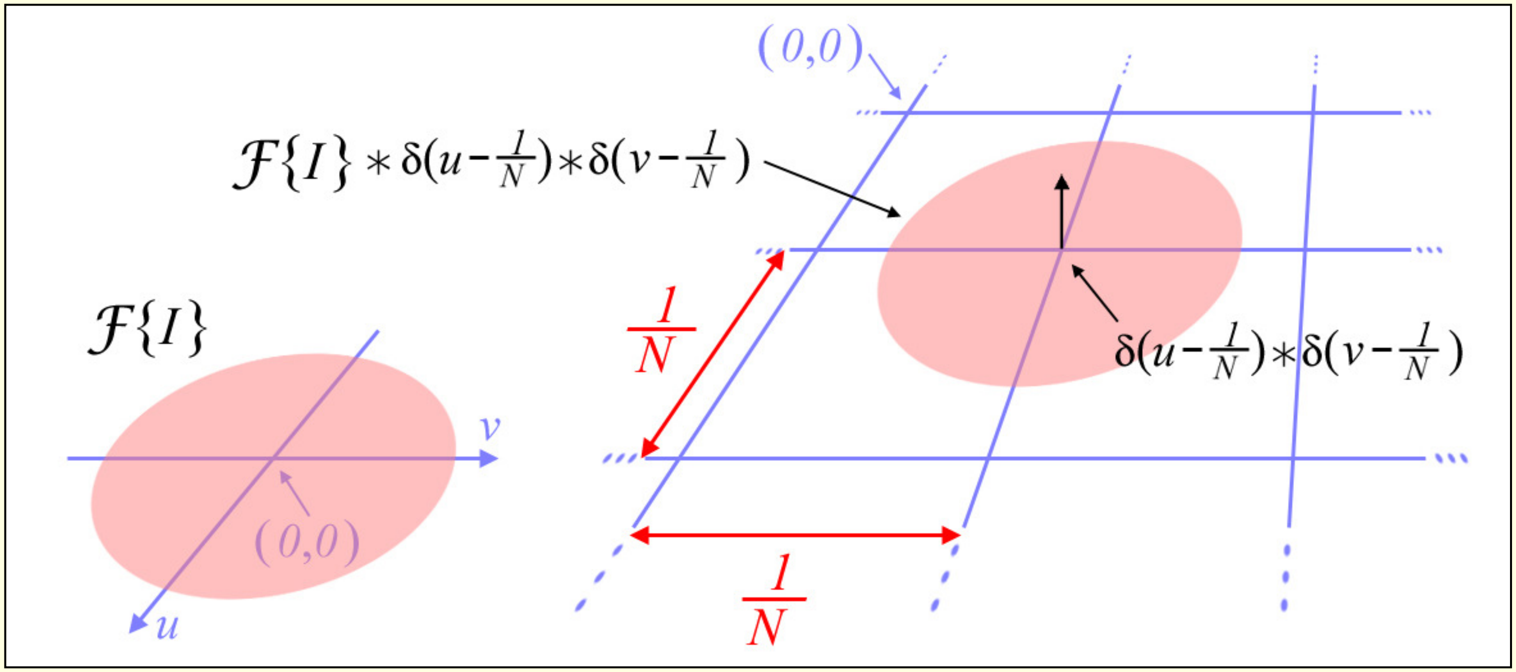
$$\mathbf{F}\{f \cdot g\} = F * G$$

$$\text{samp}_{1/N}(u,v) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \delta\left(u - \frac{j}{N}\right) \delta\left(v - \frac{k}{N}\right)$$



Aliasing

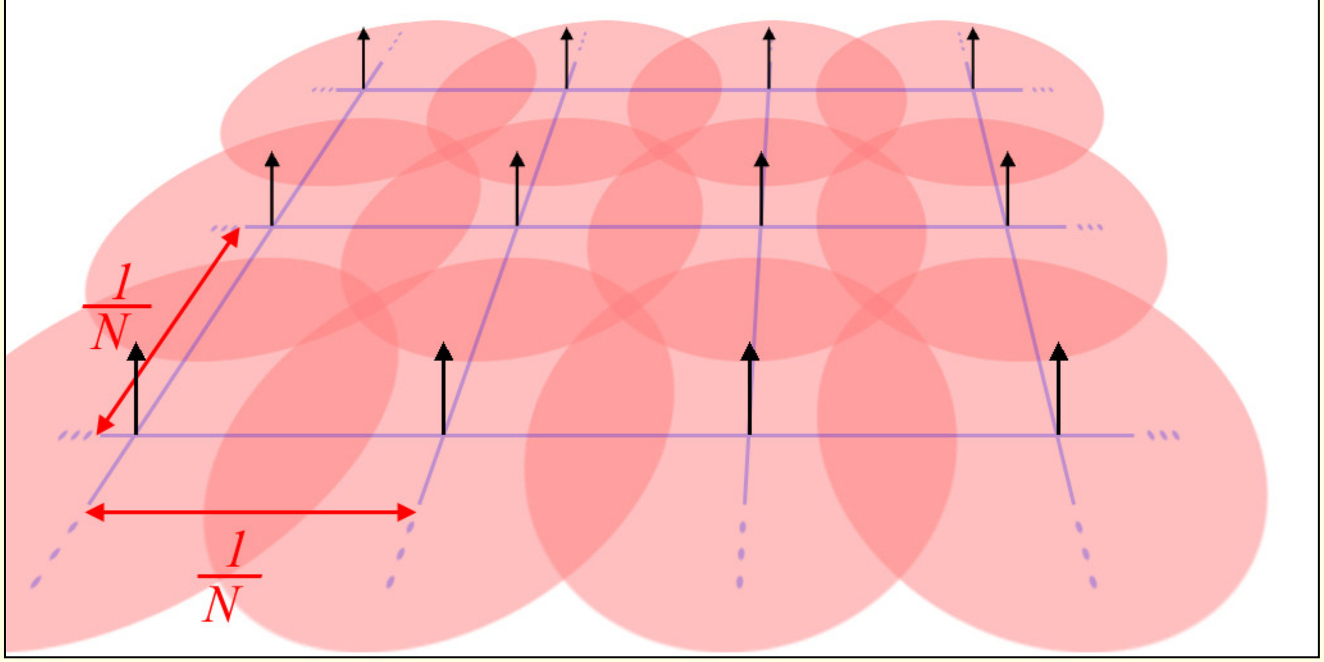
- Конволюцията на произволна функция с импулсна функция води до транслиране на функцията в позицията на импулса



Aliasing

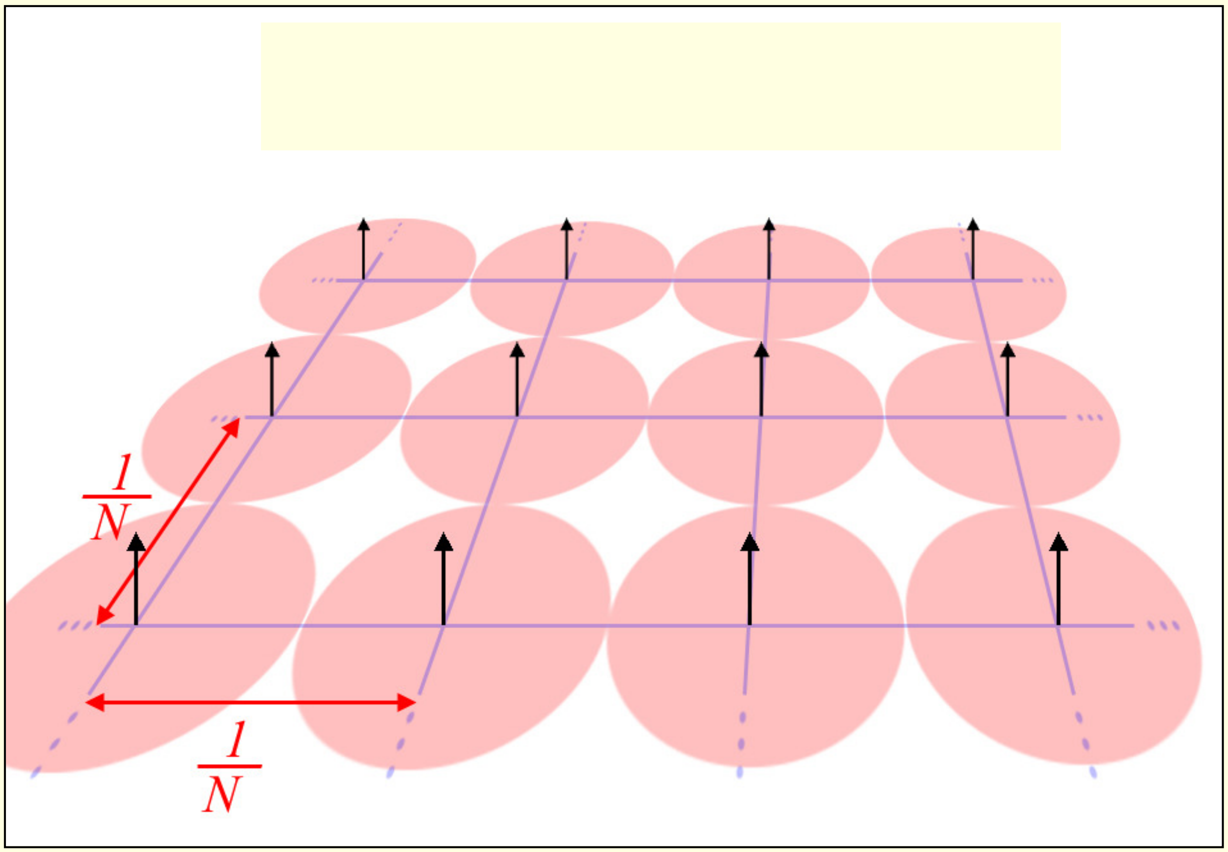
- Дискретизирането на изображението води до повтаряне на Фурие трансформацията му през интервали 1/*N*
  - при радиус по-голям от 1/(2*N*) се получава наслагване (aliasing)

$$\sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty}\mathbf{F}\{I\}(u,v)*\delta\left(u-\frac{j}{N}\right)*\delta\left(v-\frac{k}{N}\right)$$



Aliasing

- За да се избегне ефекта на aliasing
  - радиуса на Фурие трансформацията трябва да е по-малък от 1/(2*N*)



Aliasing

- За намаляване на ефекта на aliasing
  - ***филтриране преди мащабирането***

    при стъпка на дискретизация на изображението *N*

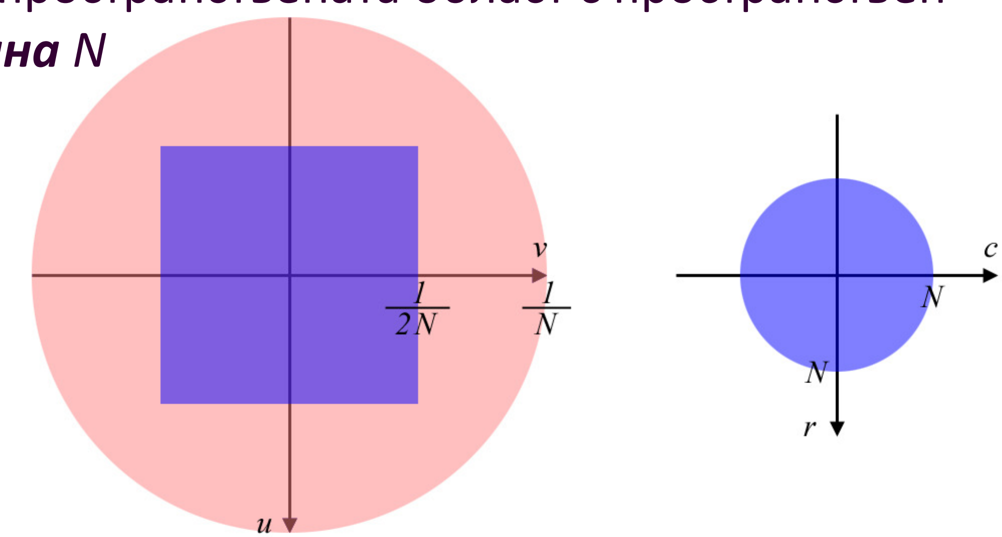
    - умножение в честотната област с $H(u,v)$, такова че

      $$H(u,v) = 0 \text{ за } u > \frac{1}{2N} \text{ или } v > \frac{1}{2N}$$

    - конволюция в пространствената област с пространствен филтър с ***ширина** N*

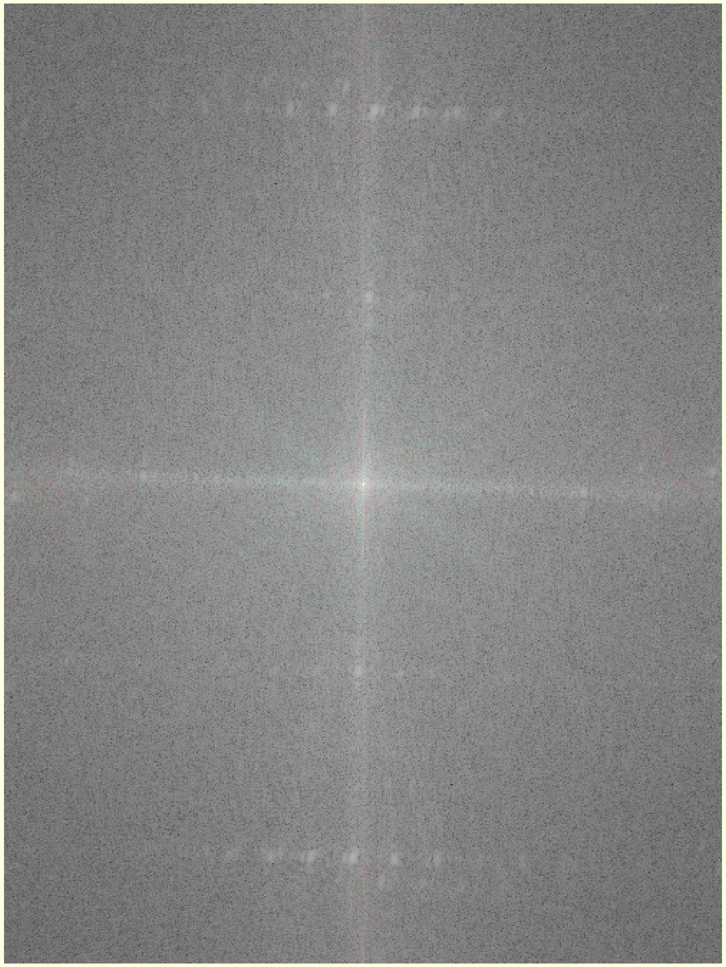
за намаляване 2 пъти размера на изображението се използва пространствен филтър с размер 4

- може да се използва филтър 3×3 или 5×5



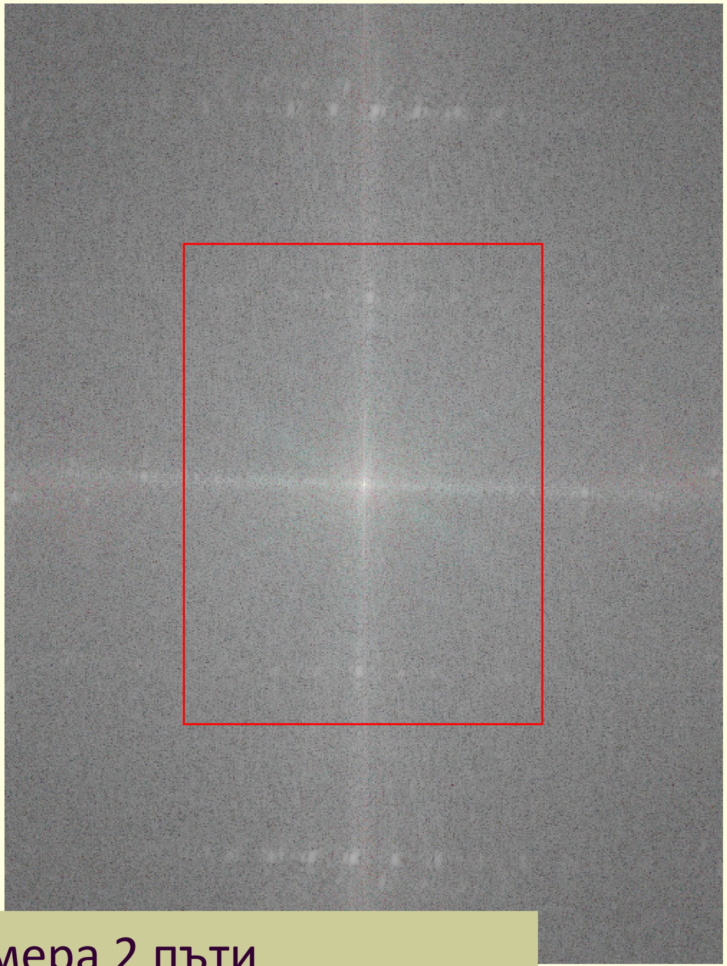
Мащабиране

оригинално изображение



спектрална функция

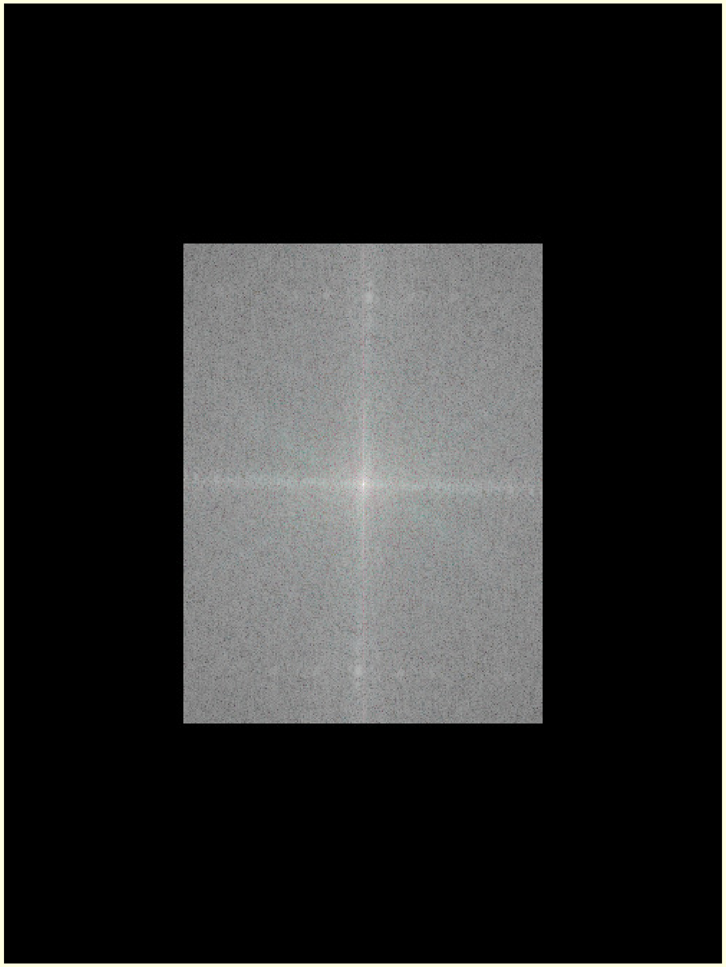
Мащабиране



за намаляване размера 2 пъти
се филтрира областта извън червения правоъгълник

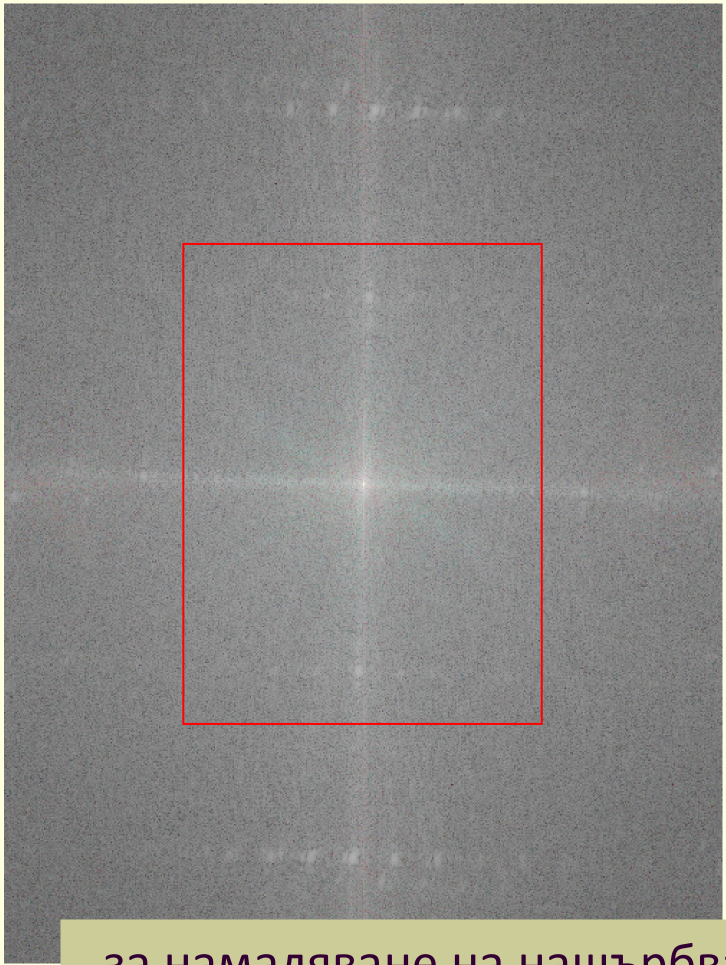
Мащабиране

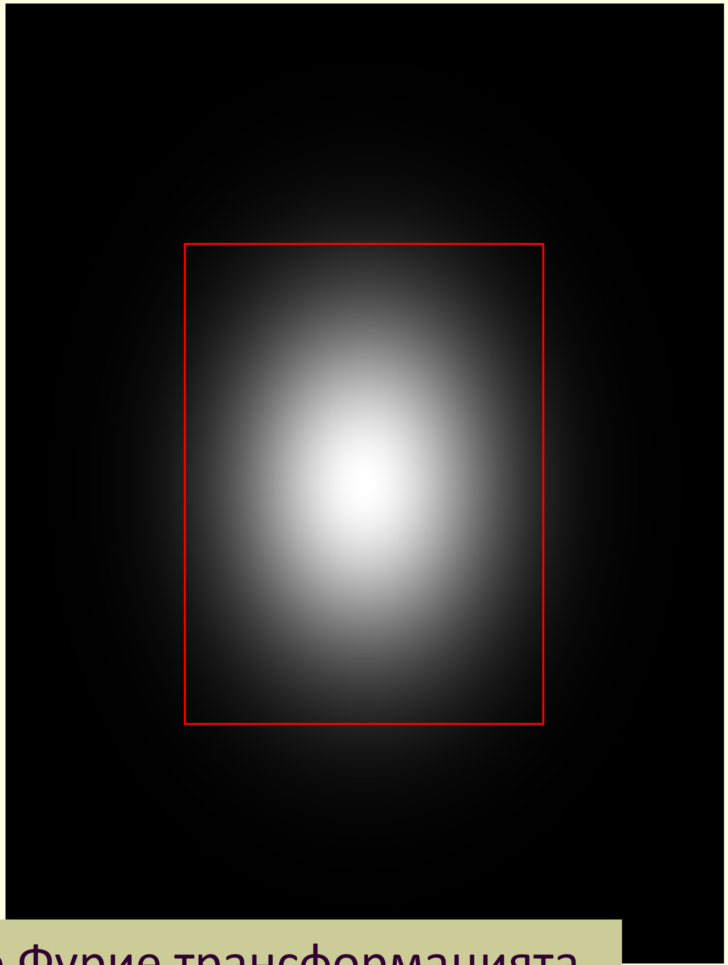
намалено изображение (zoom 2x)



идеален филтър

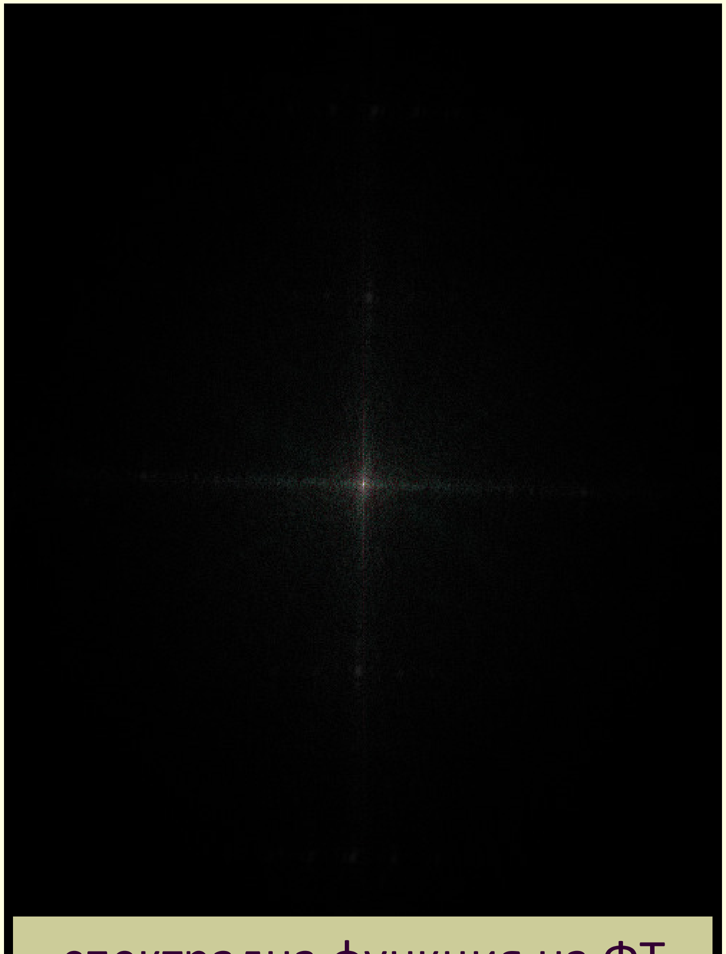
Мащабиране





за намаляване на нащърбването Фурие трансформацията се умножава с Гаусов филтър $(\sigma_v,\sigma_u)=(¼R, ¼C)$

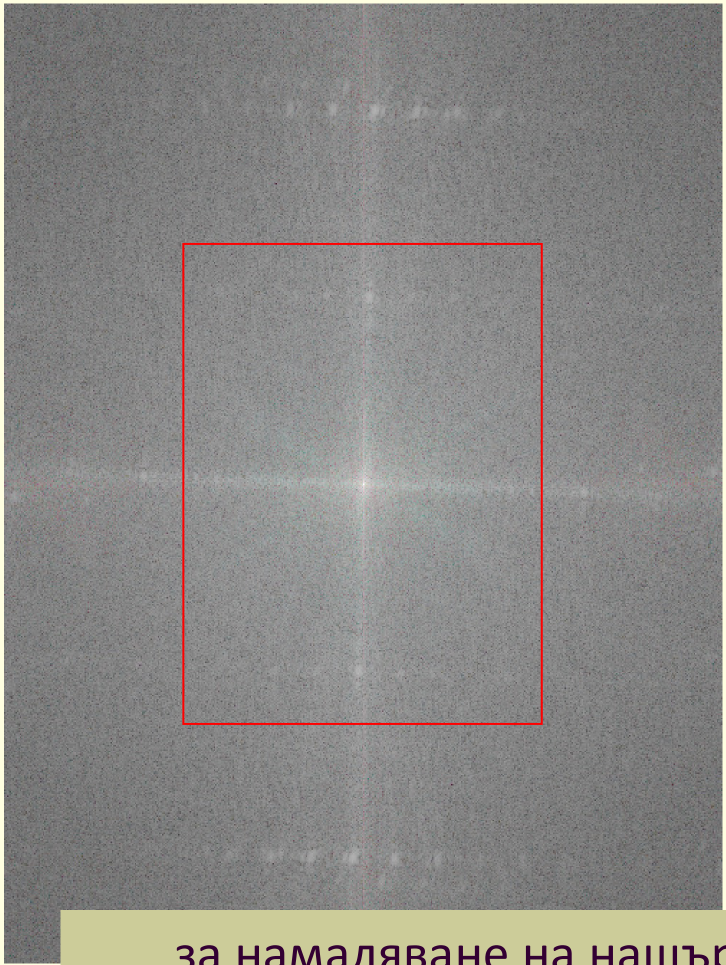
Мащабиране

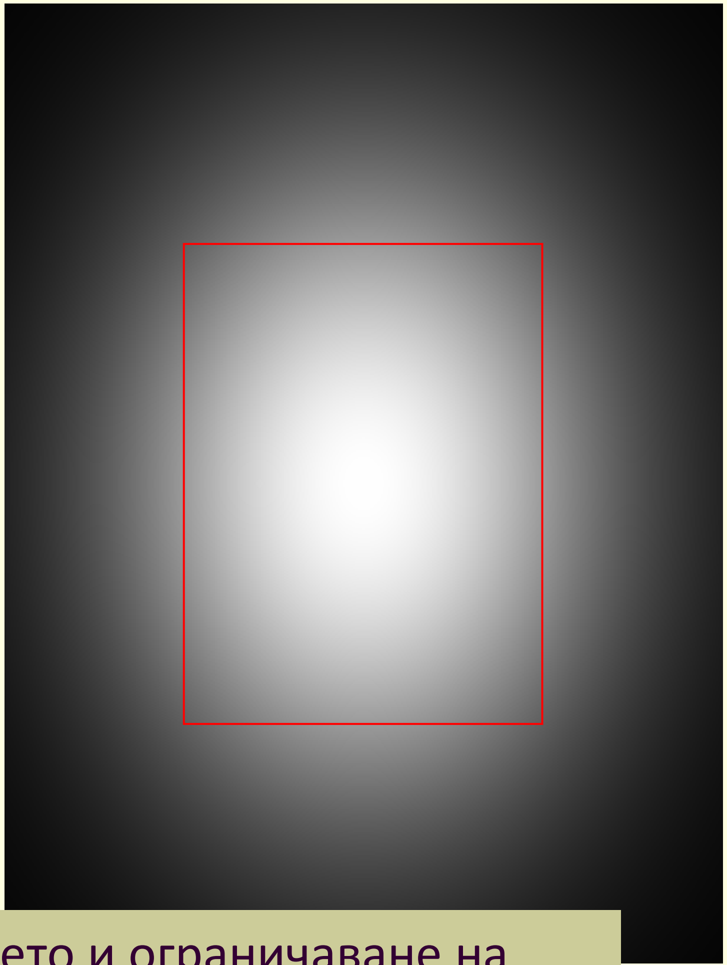


спектрална функция на ФТ
умножена с Гаусов филър

намалява нащърбването,
но изображението е размито

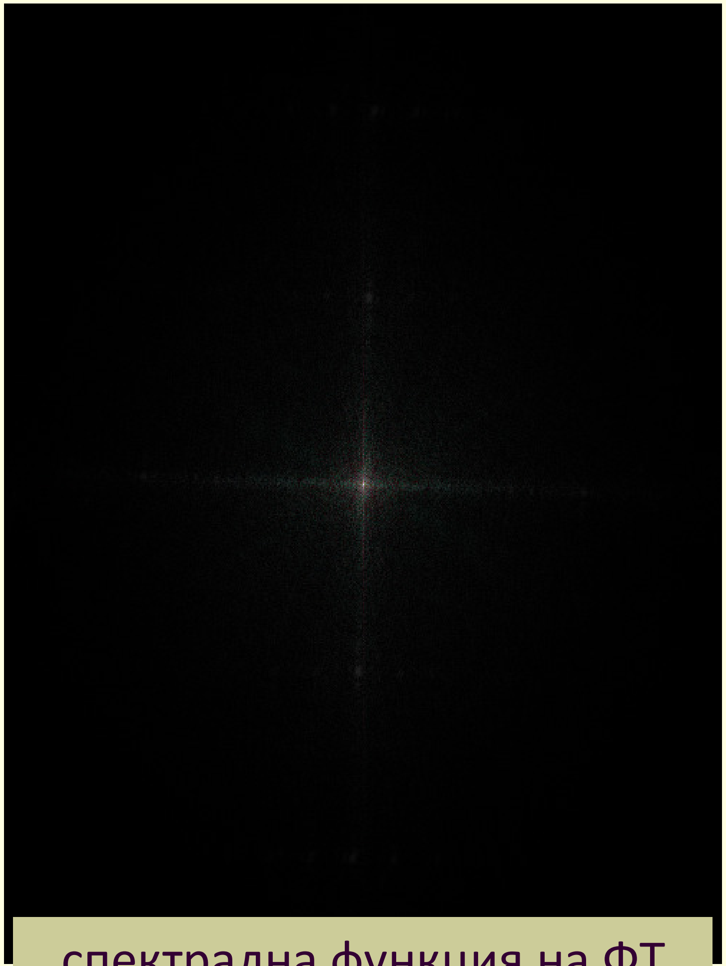
Мащабиране





за намаляване на нащърбването и ограничаване на размиването се използва Гаусов филтър $(\sigma_v,\sigma_u)=(\frac{1}{2}R, \frac{1}{2}C)$

Мащабиране



спектрална функция на ФТ
умножена с Гаусов филър

малко нащърбване
и малко размиване

Интерполация

- Три метода за интерполация на стойностите на пикселите в изображение
  - ***Най-близък съсед***
    - Nearest neighbor
  - ***Билинейна интерполация***
    - Bilinear interpolation
  - ***Бикубична интерполация***
    - Bicubic interpolation

Най-близки съседи

- ***Nearest neighbor***
- Обобщение на подхода с повторение и премахване на стойности на пиксели
  - позволява частично мащабиране
    - промяна на размера на изображение, така че да има p/q пиксели на всеки ред и p/q от редовете в оригиналното изображение
      - p и q са цели числа

Най-близки съседи

- Размери на оригиналното изображение *I*: $R' \times C'$
- Размери на мащабираното изображение *J*: $R \times C$
- Мащабиращ коефициент за редовете (входно към изходно изображение)

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

- Мащабиращ коефициент за колоните (входно към изходно изображение)

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

- За всеки пиксел (r,c) в *J* съответния пиксел от *I* е (r_f, c_f)

$$\left(r_f, c_f\right) = \left(S_R \cdot r,\ S_C \cdot c\right)$$

Най-близки съседи

- Ако $S_r \geq 0.5$ и $S_c \geq 0.5$, то $r = 1, \ldots, R$; $c = 1, \ldots, C$
- Ако $S_r < 0.5$ или $S_c < 0.5$, то $r = \lfloor 1/S_r \rfloor, \ldots, R$; $r = \lfloor 1/S_c \rfloor, \ldots, C$
- Определят се най-близките целочислени координати (r',c') на пиксел от I

$$\left(r', c'\right) = \text{round}\left(r_f, c_f\right).$$

- Стойността на всеки пиксел от мащабираното изображение се определя като

$$J\left(r, c\right) = I\left(r', c'\right).$$

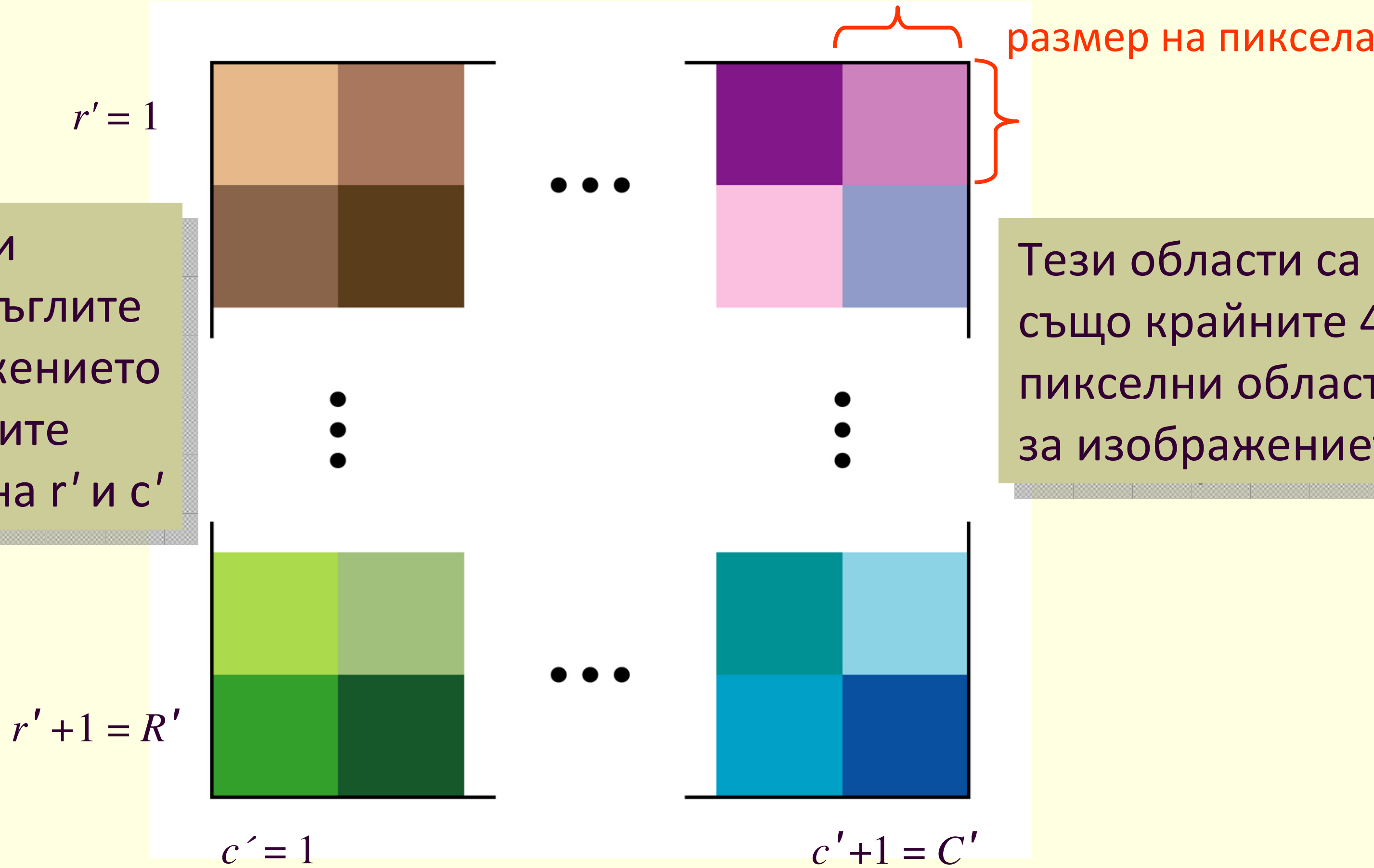
Най-близки съседи

- Локална област с размери 4×4 за всеки пиксел с координати (r_f,c_f) в оригиналното изображение I има
  - горен ляв ъгъл $(r',c') = \lfloor (r_f,c_f) \rfloor$
  - долен десен ъгъл $(r'+1,c'+1)$
- За всеки пиксел с координати (r_f,c_f)

  (1) нито r', нито c' не могат да бъдат по-малки от 1

  (2) r'+1 не може да бъде по-голямо от R' и c'+1 не може да бъде по-голямо от C'
- Ако множеството от всички индекси $\{(r',c')\}$ не удовлетворяват (1) или (2), то индексите се променят за да бъдат изпълнени тези условия

Най-близки съседи

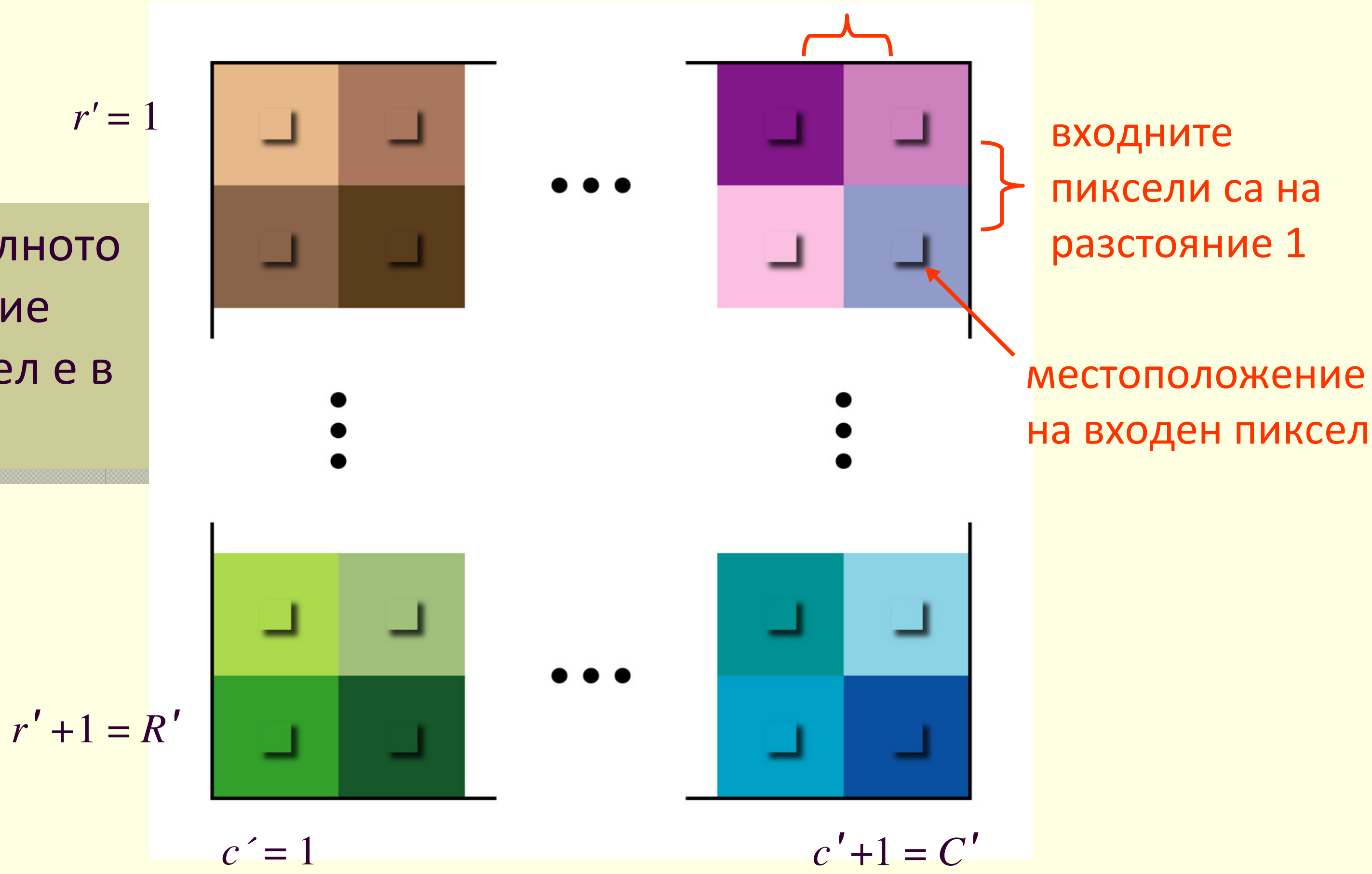


Тези четири области са ъглите на изображението – екстремните стойности на r′ и c′

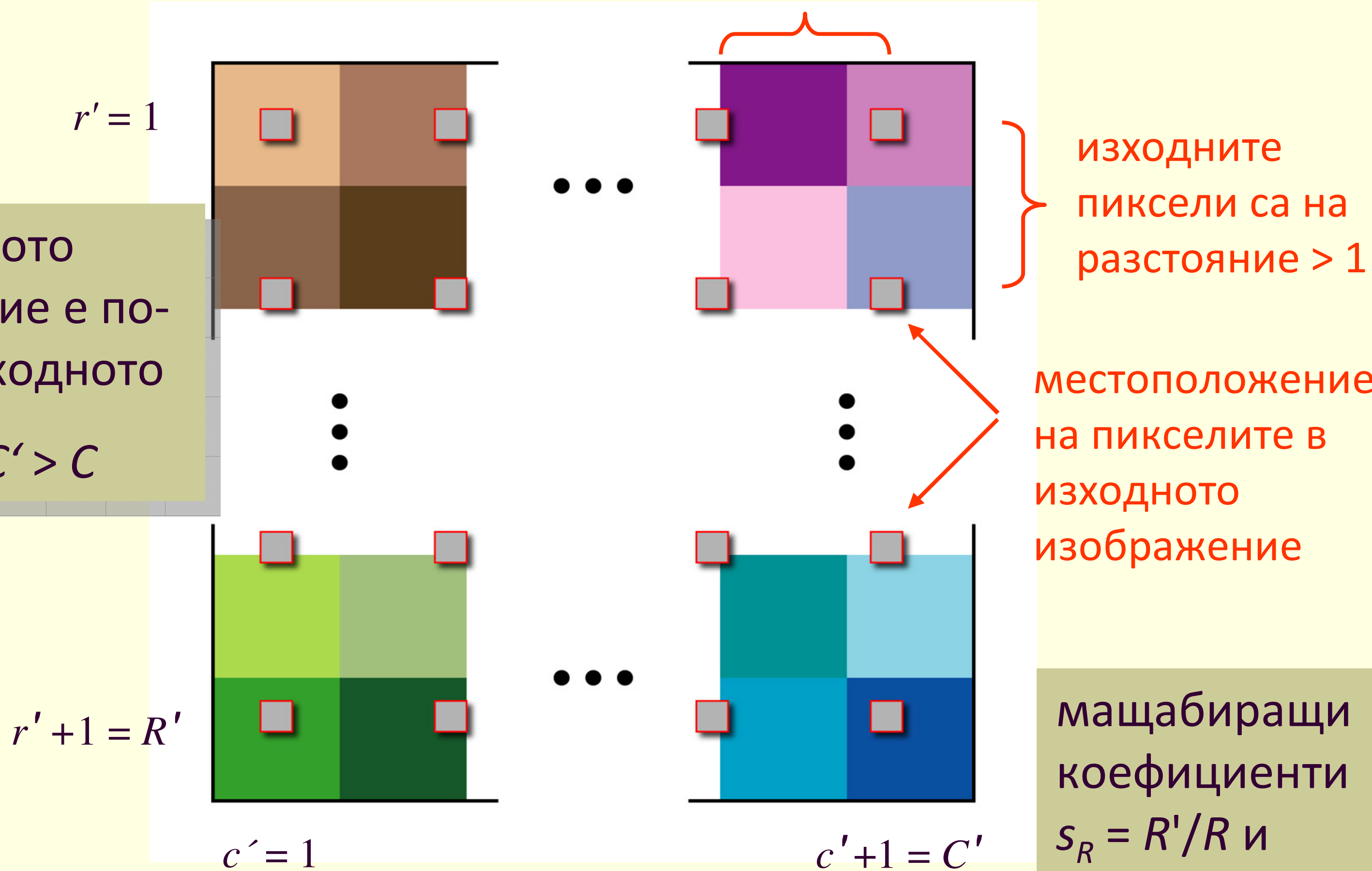
Тези области са също крайните 4-пикселни области за изображението

Най-близки съседи

За оригиналното изображение всеки пиксел е в центъра



Най-близки съседи



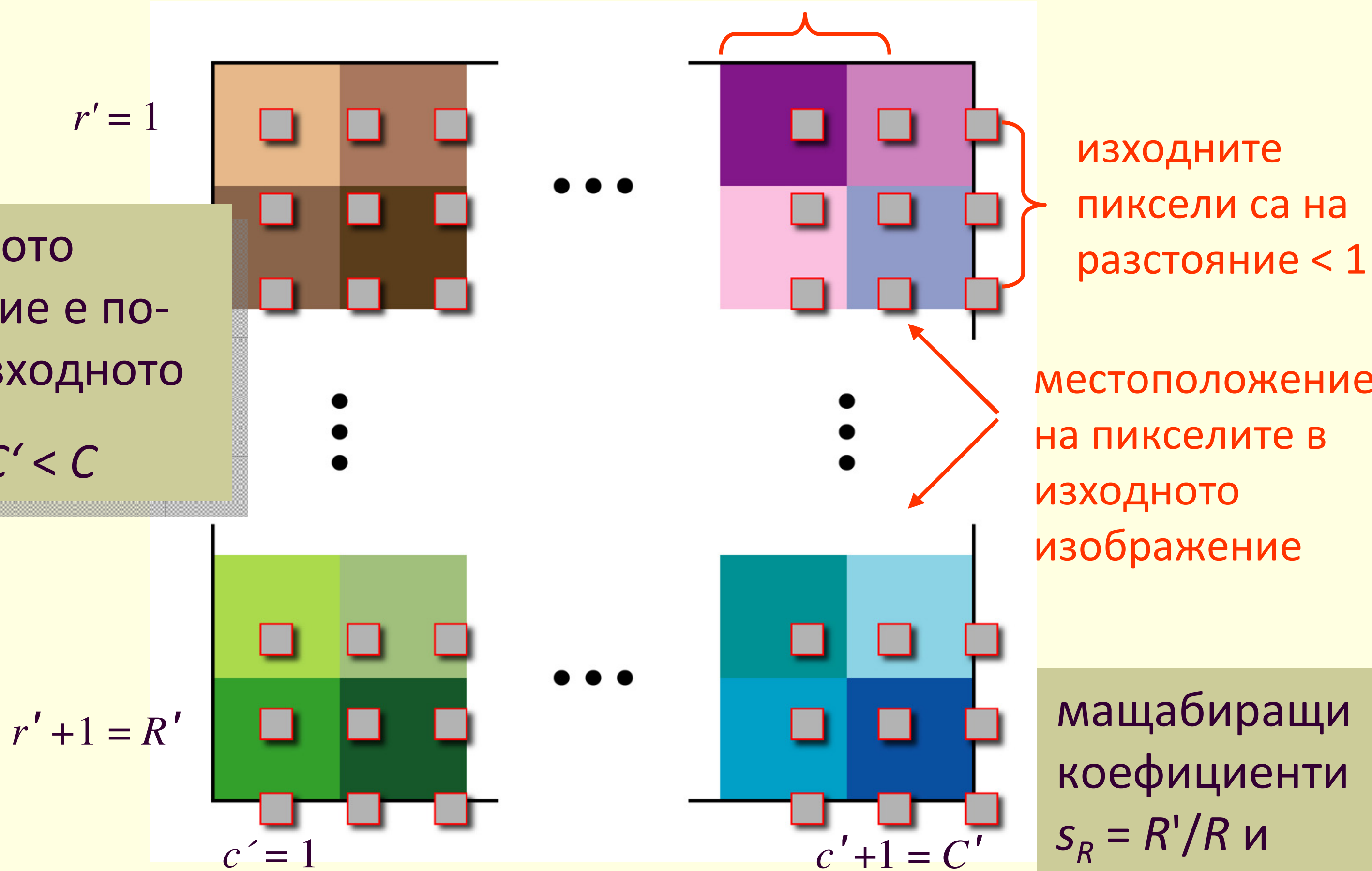
Ако изходното изображение е по-малко от входното то $R' > R$ и $C' > C$

мащабиращи коефициенти
$s_R = R'/R$ и
$s_C = C'/C$

Най-близки съседи

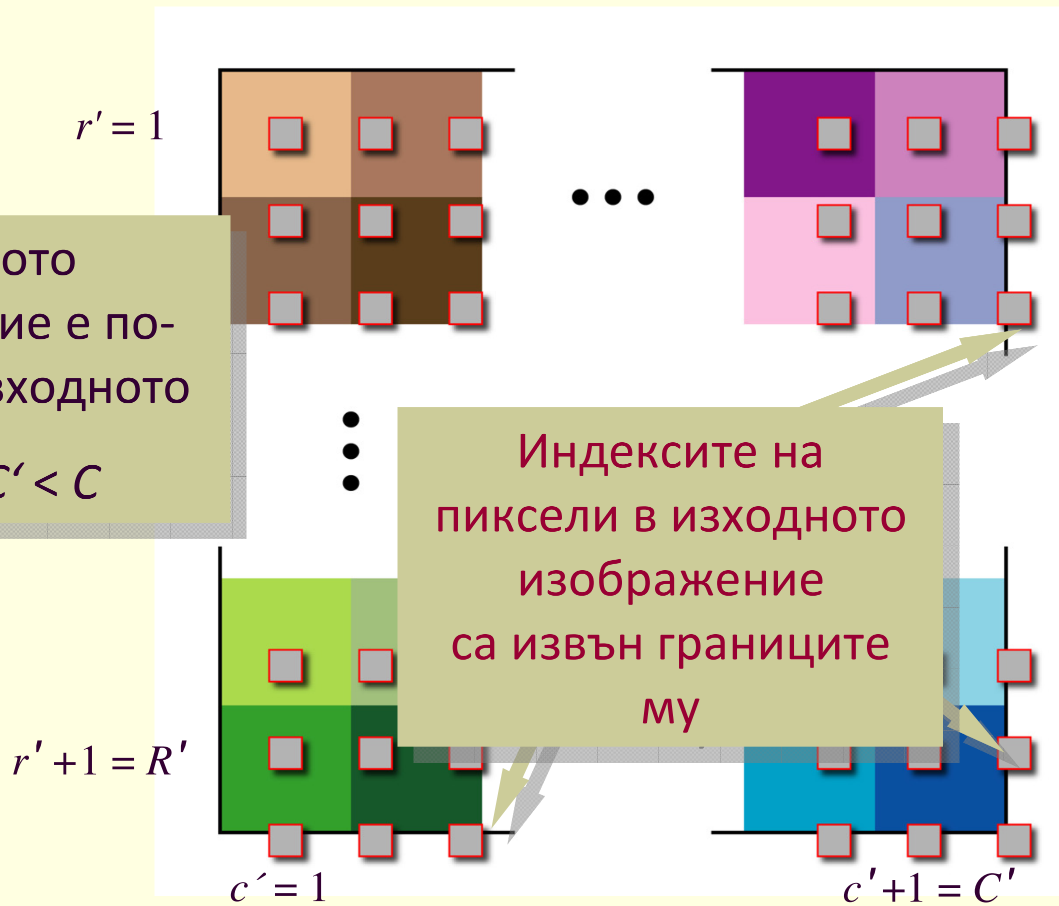
Ако изходното изображение е по-голямо от входното

то $R' < R$ и $C' < C$



мащабиращи коефициенти
$s_R = R'/R$ и
$s_C = C'/C$

Най-близки съседи



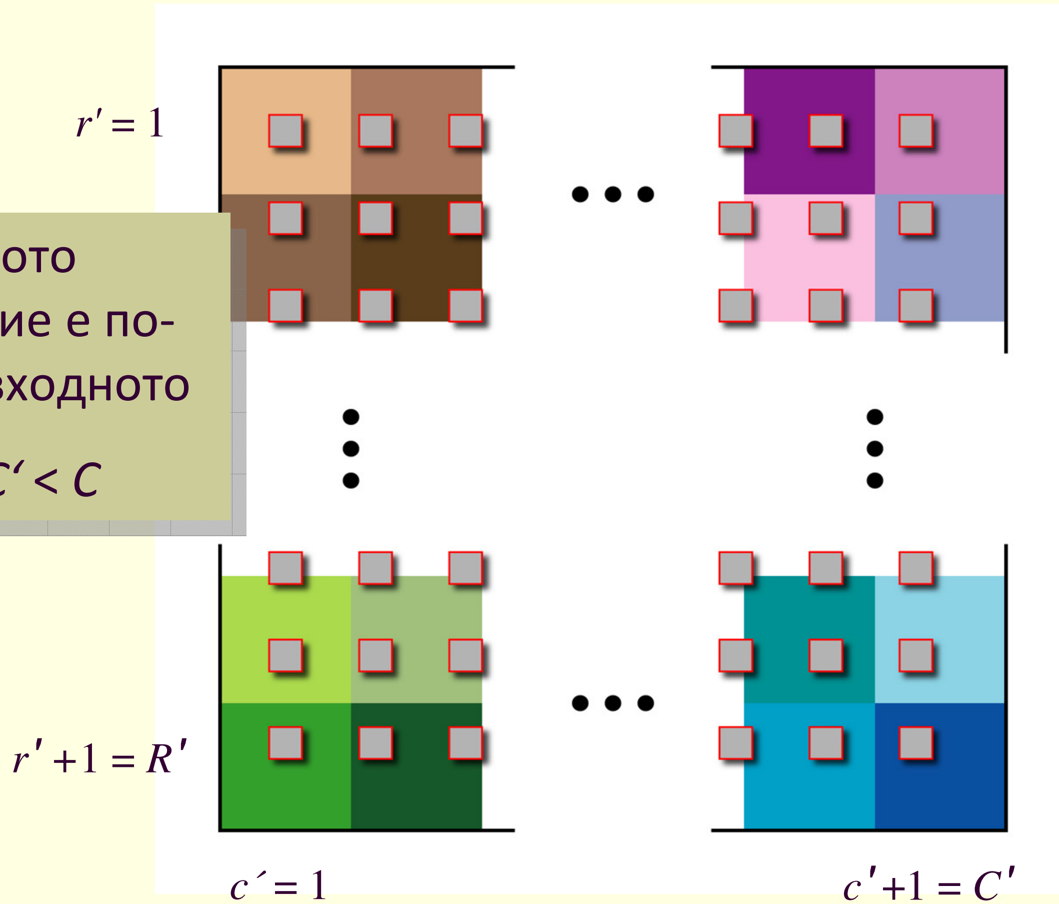
Ако изходното изображение е по-голямо от входното

то $R' < R$ и $C' < C$

Индексите на пиксели в изходното изображение са извън границите му

мащабиращи коефициенти
$s_R = R'/R$ и
$s_C = C'/C$

Най-близки съседи



Ако изходното изображение е по-голямо от входното

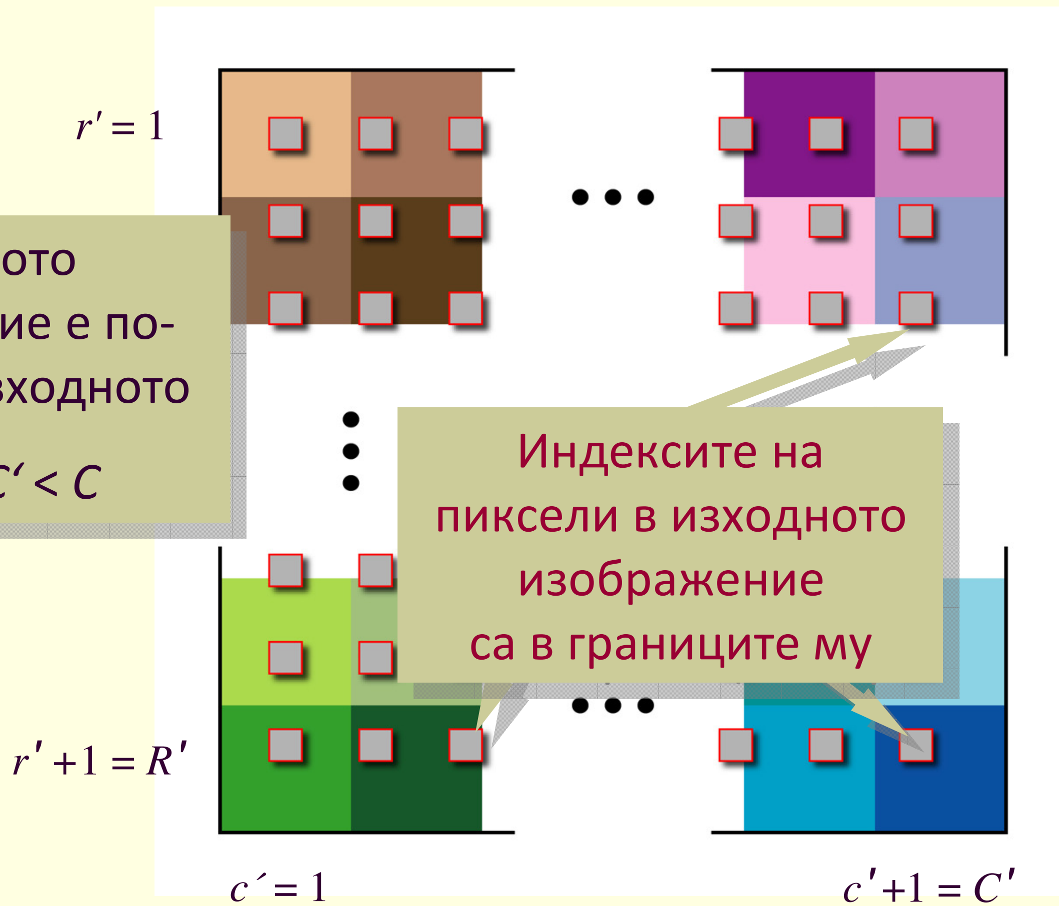
то $R' < R$ и $C' < C$

Използват се мащабиращи коефициенти
$s_R = (R'-1)/R$ и
$s_C = (C'-1)/C$

Най-близки съседи

Ако изходното изображение е по-голямо от входното

то $R' < R$ и $C' < C$

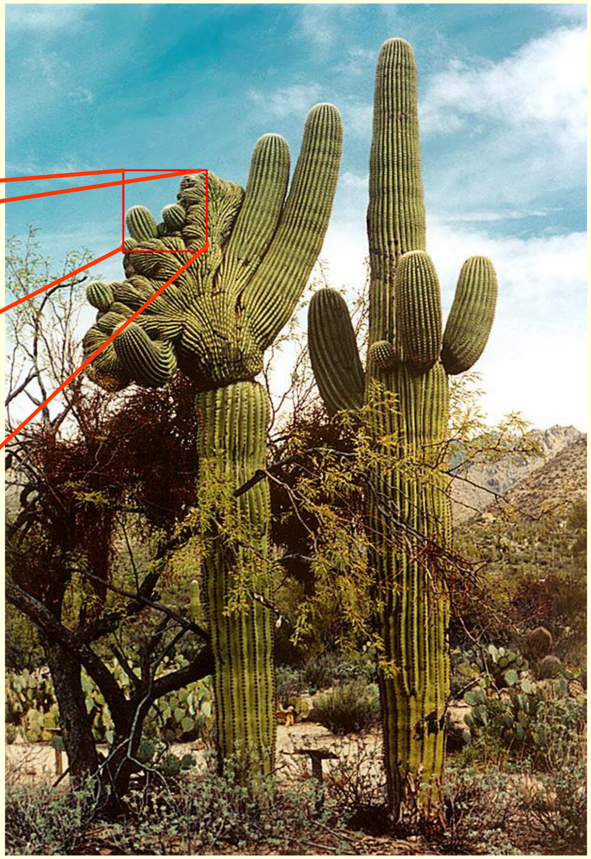


Индексите на пиксели в изходното изображение
са в границите му

Използват се мащабиращи коефициенти
$s_R = (R'-1)/R$ и
$s_C = (C'-1)/C$

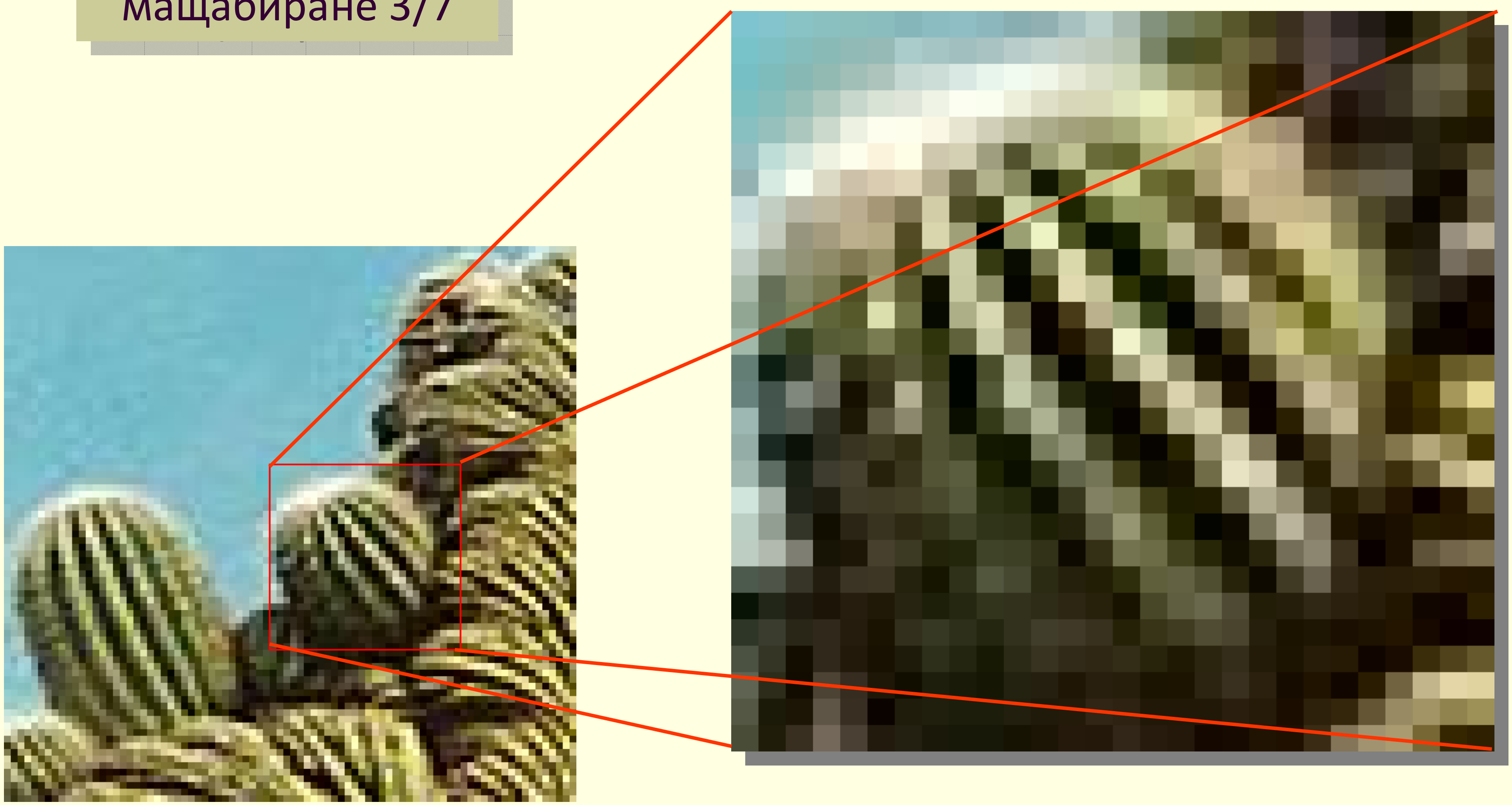
Най-близки съседи

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



Най-близки съседи

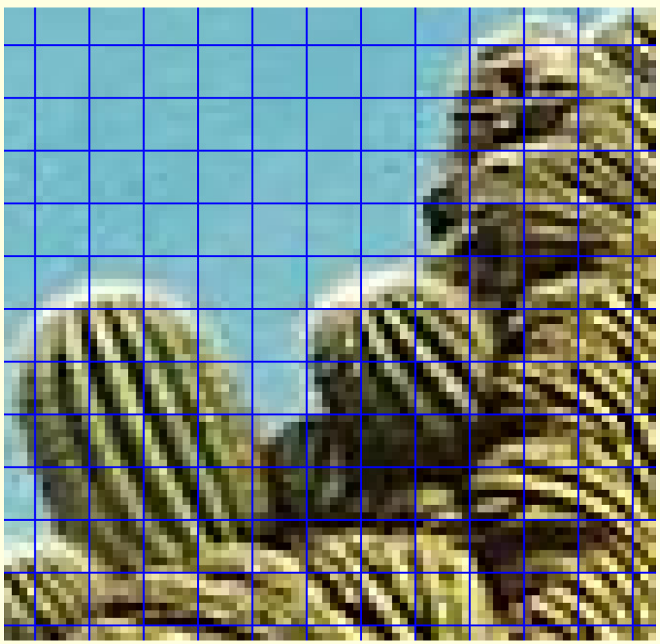
мащабиране 3/7

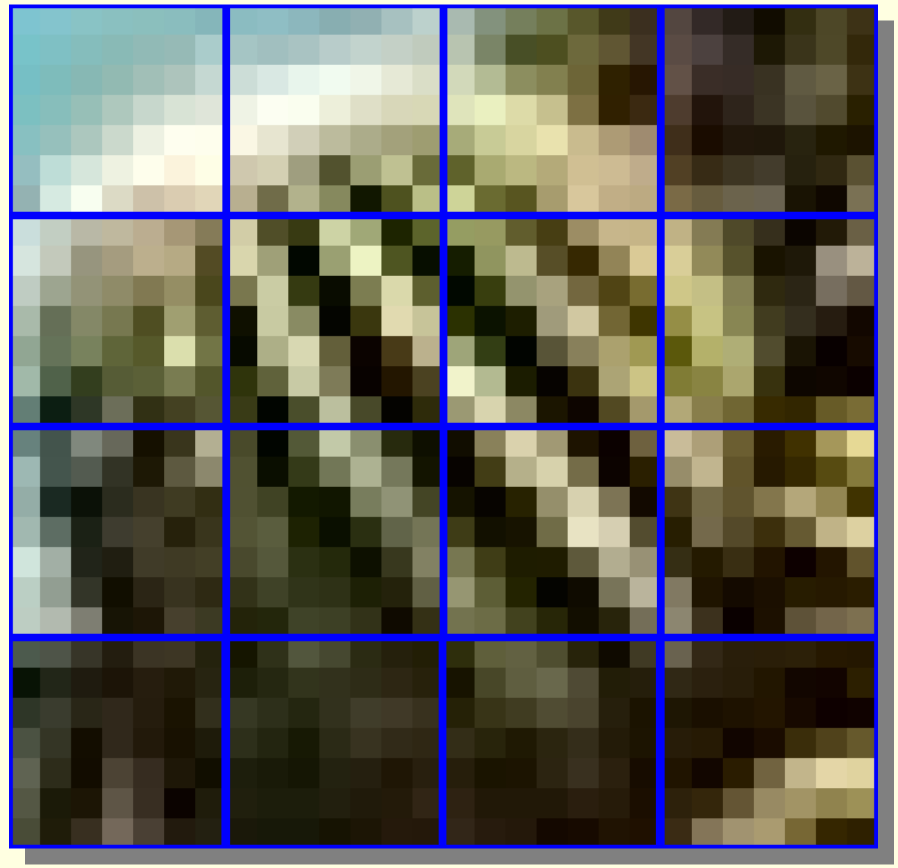


Най-близки съседи

мащабиране 3/7

В син цвят:
7x7 пиксела





Най-близки съседи

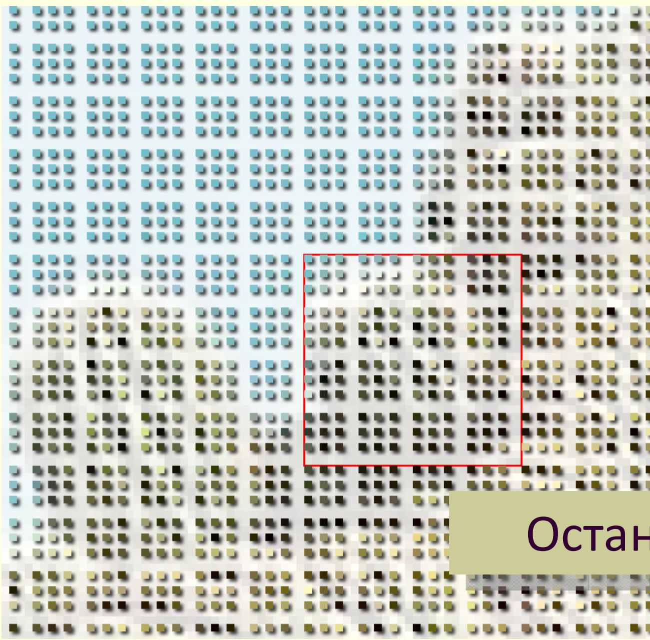
мащабиране 3/7

В жълт цвят:
3 пиксела за всеки 7 реда,
3 пиксела за всеки 7 колони

Запазват се само пикселите в жълто

Най-близки съседи

мащабиране 3/7



Останалите се игнорират

Най-близки съседи

мащабиране 3/7

Копират се в ново изображение

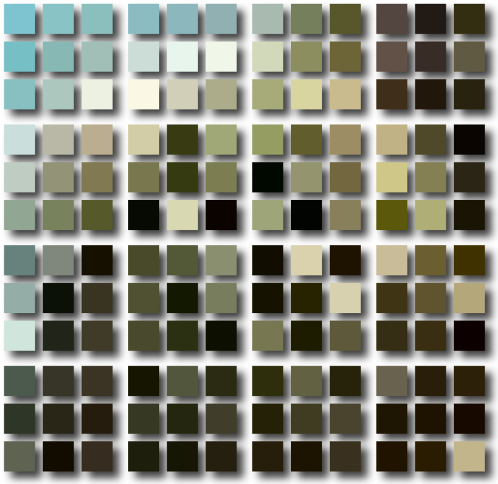
Най-близки съседи

мащабиране 3/7

Копират се в ново изображение

Най-близки съседи

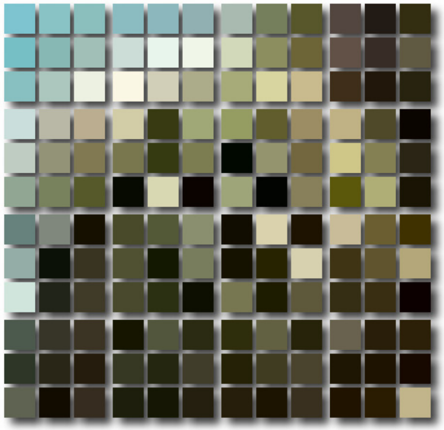
мащабиране 3/7



Копират се в ново изображение

Най-близки съседи

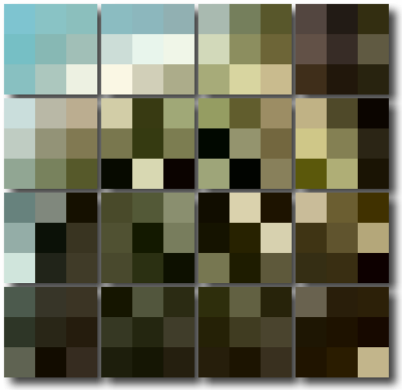
мащабиране 3/7



Копират се в ново изображение

Най-близки съседи

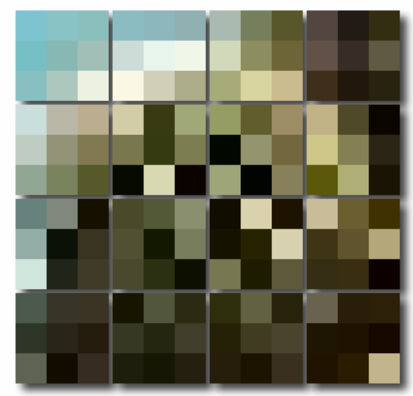
мащабиране 3/7



Копират се в ново изображение с размери 3/7 от тези на оригиналното

Най-близки съседи

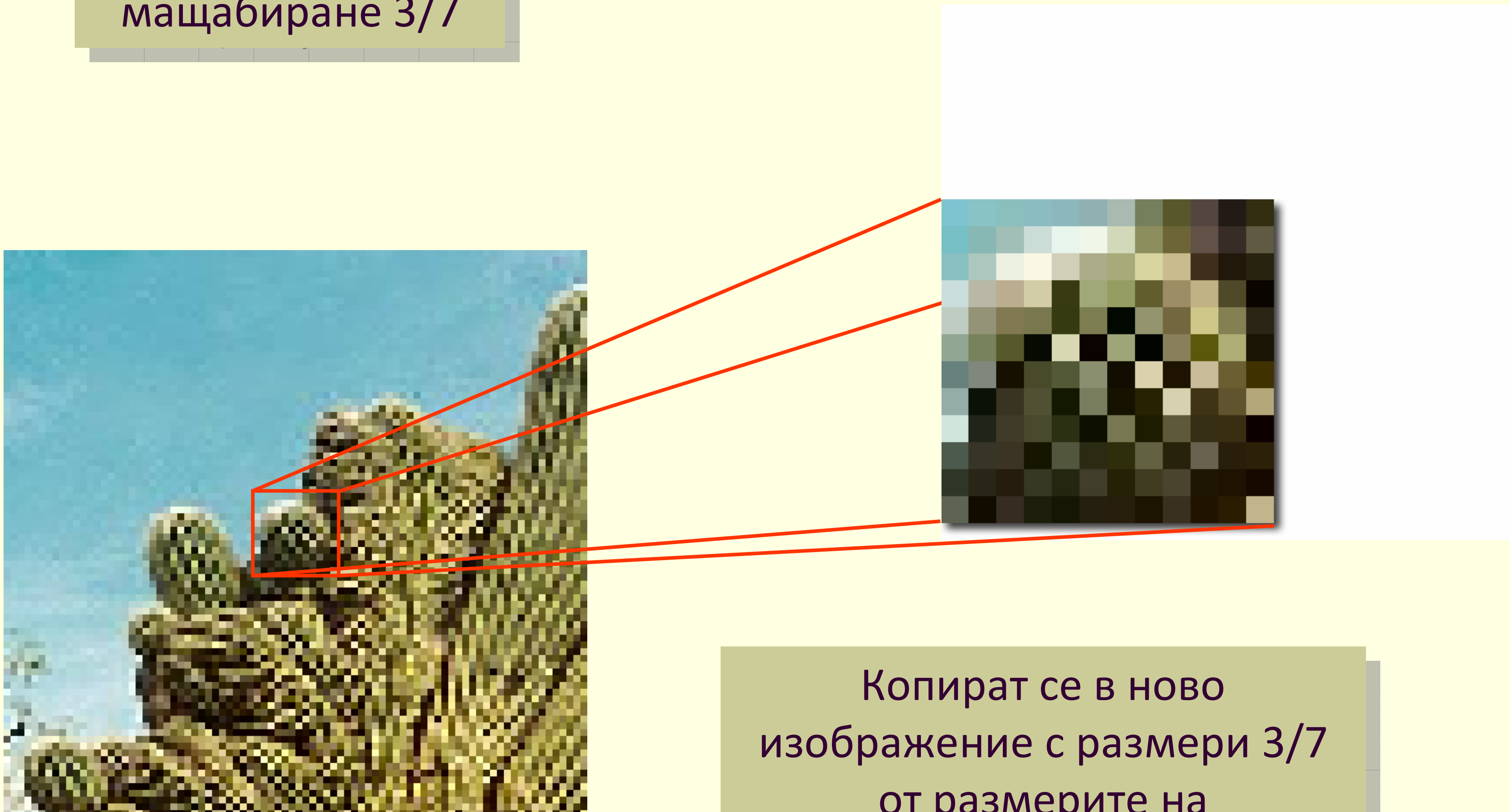
мащабиране 3/7



Копират се в ново изображение

Най-близки съседи

мащабиране 3/7



Копират се в ново изображение с размери 3/7 от размерите на оригиналното

Най-близки съседи

мащабиране към размери 3/7 от оригиналното изображение

Най-близки съседи

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

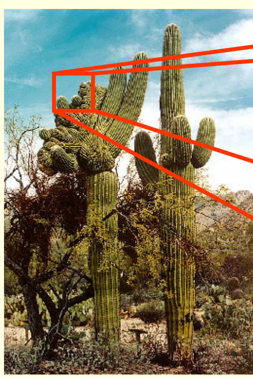
Оригинални пиксели в мащабираното изображение

Мащабирано 7/3 изображение

Най-близки съседи

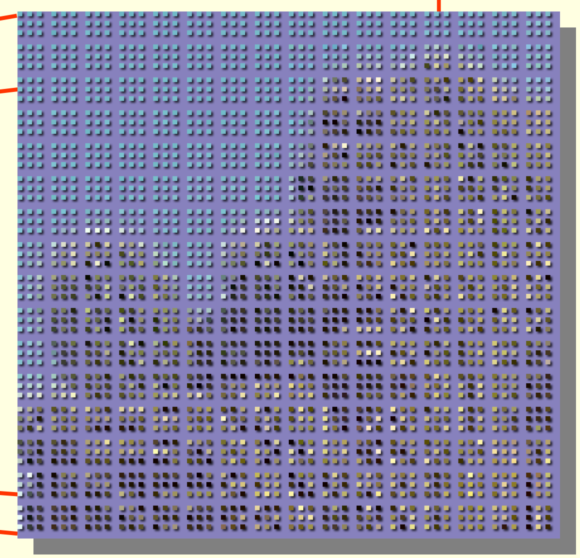
мащабиране 7/3

Всяка област с размери 3×3 от оригиналното изображение се разпростира в област с размери 7×7 в мащабираното изображение



оригинално изображение

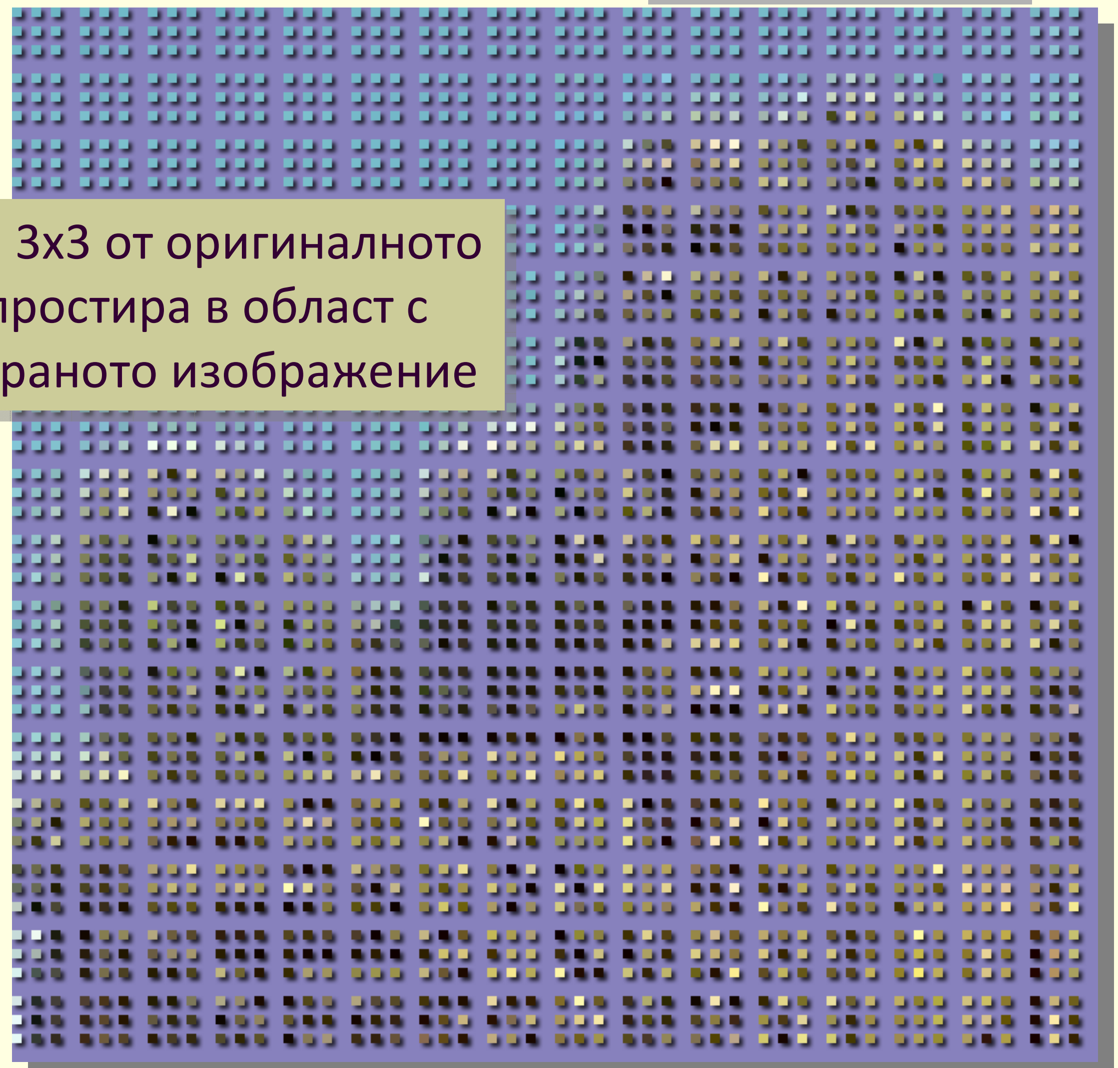
детайл



Най-близки съседи

мащабиране 7/3

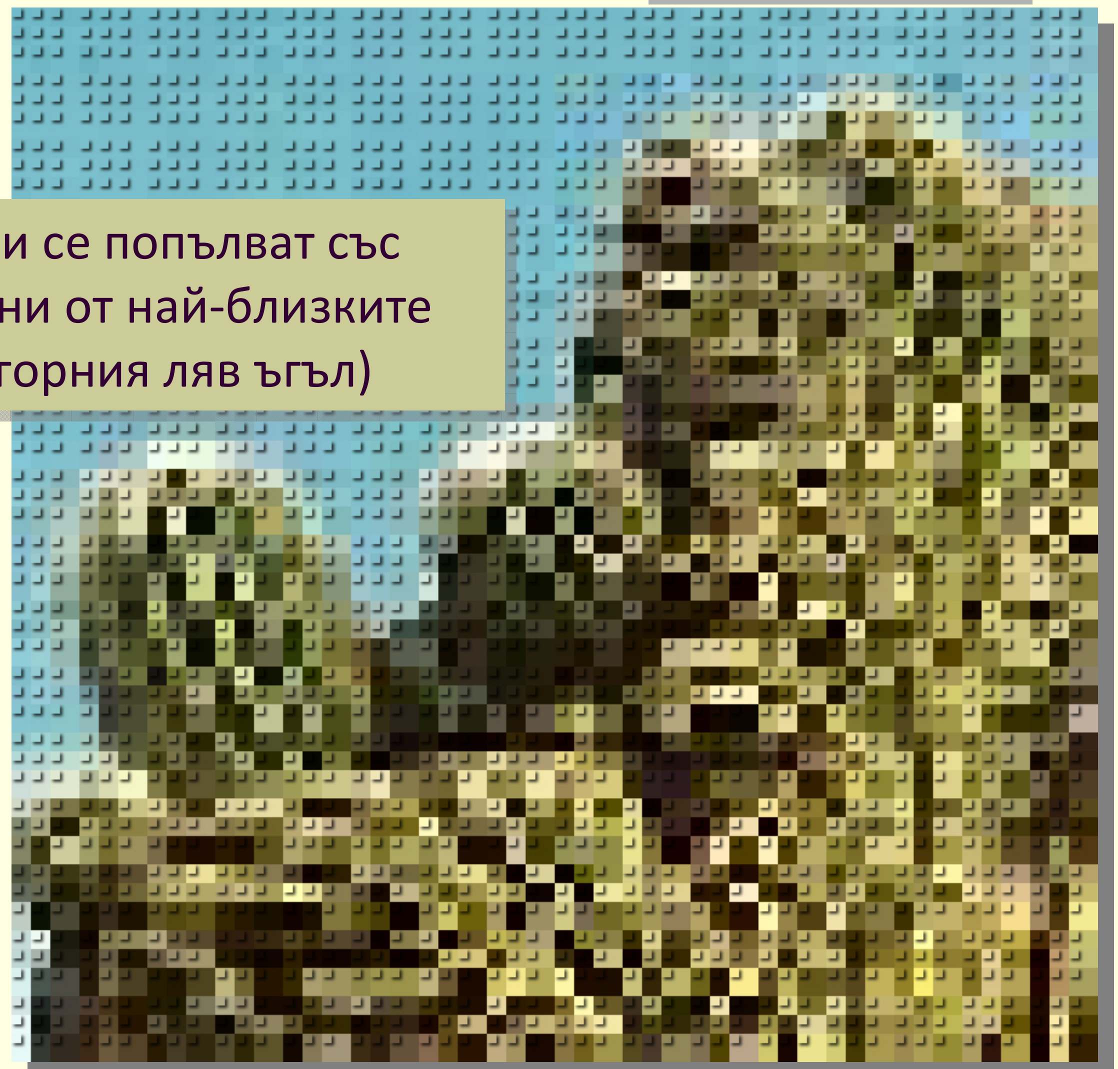
Всяка област с размери 3x3 от оригиналното изображение се разпростира в област с размери 7×7 в мащабираното изображение



Най-близки съседи

мащабиране 7/3

Липсващите пиксели се попълват със стойности, определени от най-близките съседни пиксели (горния ляв ъгъл)



Най-близки съседи

мащабиране 7/3

Липсващите пиксели се попълват със стойности, определени от най-близките съседни пиксели (горния ляв ъгъл)

Най-близки съседи

мащабиране 7/3

оригинално
изображение

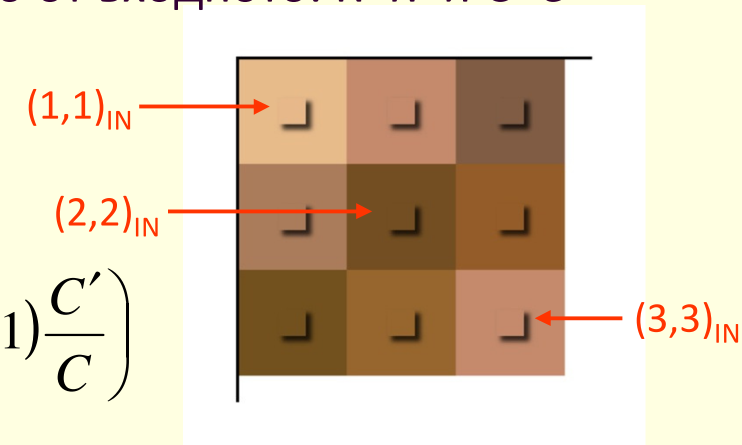
мащабирано 7/3
изображение

Най-близки съседи

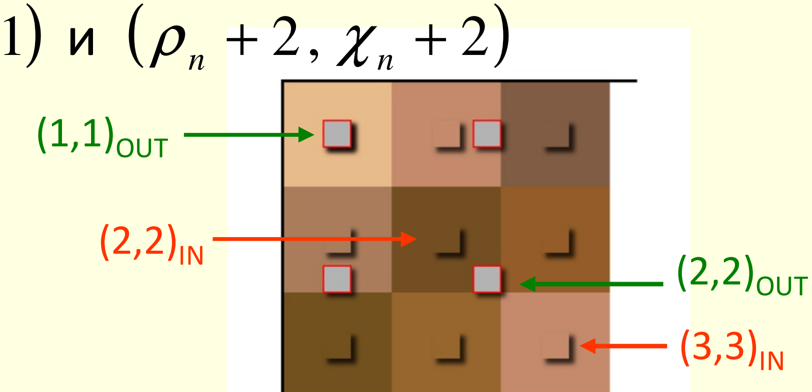
- Ако изходното изображение е по-малко от входното: R<*R′* и C<*C′*
- то $(1,1)_{\text{OUT}} \leftrightarrow (1,1)_{\text{IN}}$

$$(2,2)_{\text{OUT}} \leftrightarrow (1,1)_{\text{IN}} + \left(\frac{R'}{R}, \frac{C'}{C}\right)$$

$$(n,n)_{\text{OUT}} \leftrightarrow (1,1)_{\text{IN}} + \left((n-1)\frac{R'}{R}, (n-1)\frac{C'}{C}\right)$$



- $(n,n)_{\text{OUT}}$ лежи между $(\rho_n + 1, \chi_n + 1)$ и $(\rho_n + 2, \chi_n + 2)$



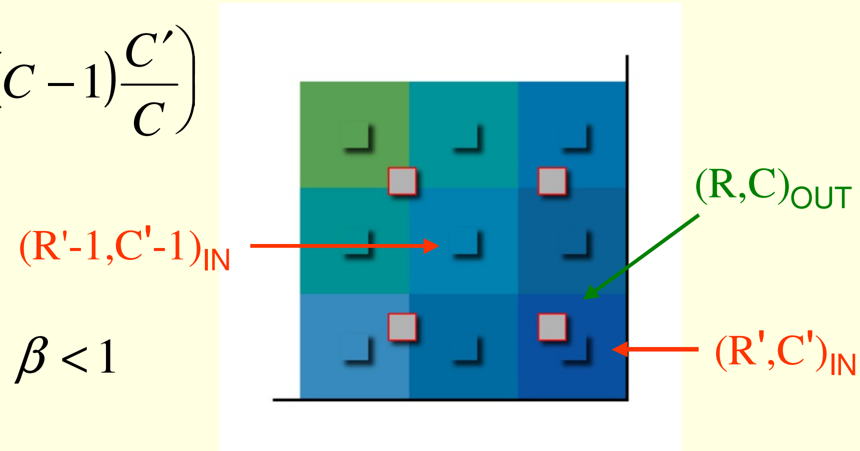
- където

$$\rho_n = \left\lfloor (n-1)\frac{R'}{R} \right\rfloor \text{ и } \chi_n = \left\lfloor (n-1)\frac{C'}{C} \right\rfloor$$

$(2,2)_{\text{OUT}}$ лежи между $(2,2)_{\text{IN}}$ и $(3,3)_{\text{IN}}$ тъй като R′/R > 1 & C′/C>1

Най-близки съседи

- Ако изходното изображение е по-голямо от входното: R>*R'* и C>*C'*
- то $(R,C)_{\mathrm{OUT}} \leftrightarrow (1,1)_{\mathrm{IN}} + \left((R-1)\frac{R'}{R},(C-1)\frac{C'}{C}\right)$



- но

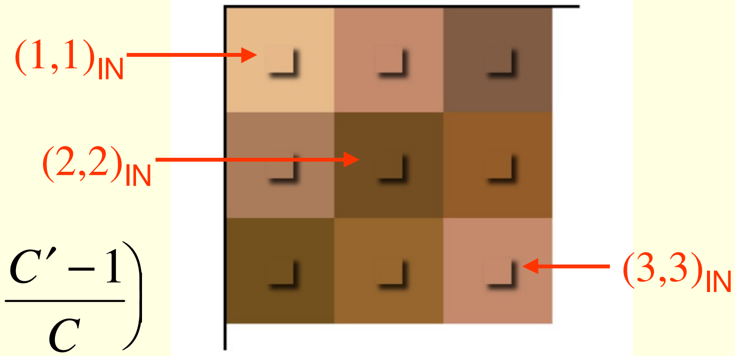
$R = \alpha R'$ и $C = \beta C'$ където $\alpha < 1$ и $\beta < 1$

$$\rho_R = \left\lfloor (R-1)\frac{R'}{R} \right\rfloor = \left\lfloor (\alpha R'-1)\frac{R'}{\alpha R'} \right\rfloor = \left\lfloor \frac{1}{\alpha}(\alpha R'-1) \right\rfloor = \left\lfloor R' - \frac{1}{\alpha} \right\rfloor = R'-2.$$

- аналогично $\chi_C = C' - 2$
- следователно $(R,C)_{\mathrm{OUT}}$ лежи между $(R'-1, C'-1)_{\mathrm{IN}}$ и $(R', C')_{\mathrm{IN}}$

Най-близки съседи

- Ако изходното изображение е по-голямо от входното: R>R' и C>C'

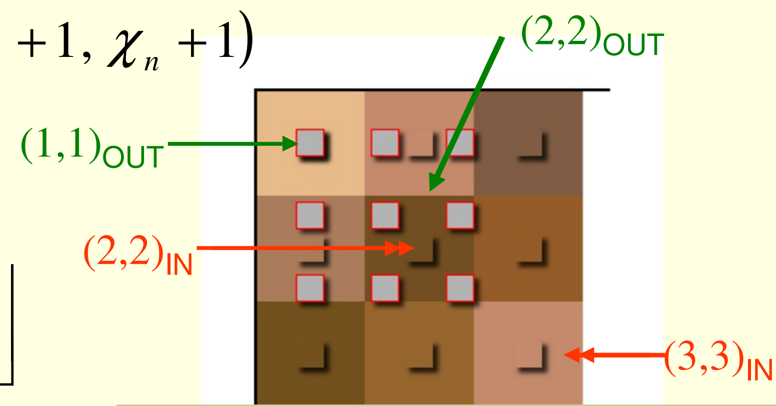


- то $(1,1)_{\text{OUT}} \leftrightarrow (1,1)_{\text{IN}}$

$$(2,2)_{\text{OUT}} \leftrightarrow (1,1)_{\text{IN}} + \left(\frac{R'-1}{R}, \frac{C'-1}{C}\right)$$

$$(n,n)_{\text{OUT}} \leftrightarrow (1,1)_{\text{IN}} + \left((n-1)\frac{R'-1}{R}, (n-1)\frac{C'-1}{C}\right)$$

- $(n,n)_{\text{OUT}}$ лежи между (ρ_n, χ_n) и (ρ_n+1, χ_n+1)



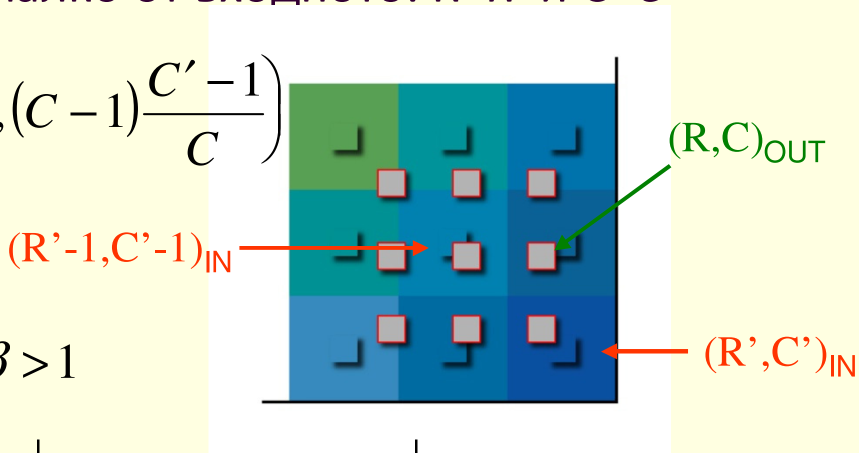
- където

$$\rho_n = \left\lfloor (n-1)\frac{R'-1}{R} \right\rfloor \text{ и } \chi_n = \left\lfloor (n-1)\frac{C'-1}{C} \right\rfloor$$

$(2,2)_{\text{OUT}}$ лежи между $(1,1)_{\text{IN}}$ и $(2,2)_{\text{IN}}$
тъй като (R’-1)/R<1

Най-близки съседи

- Ако изходното изображение е по-малко от входното: R<R' и C<C'
- то $(R,C)_{\text{OUT}} \leftrightarrow (1,1)_{\text{IN}} + \left((R-1)\frac{R'-1}{R},(C-1)\frac{C'-1}{C}\right)$



- но

$R=\alpha R'$ и $C=\beta C'$ където $\alpha>1$ и $\beta>1$

$$\rho_R = \left\lfloor (R-1)\frac{R'-1}{R} \right\rfloor = \left\lfloor (\alpha R'-1)\frac{R'-1}{\alpha R'} \right\rfloor = \left\lfloor R'-1-\frac{1}{\alpha}+\frac{1}{\alpha R'} \right\rfloor = R'-2.$$

- аналогично $\chi_C = C'-2$
- следователно $(R,C)_{\text{OUT}}$ лежи между $(R'-1,C'-1)_{\text{IN}}$ и $(R',C')_{\text{IN}}$