IMAGE SAMPLING AND IMAGE QUANTIZATION

1. Introduction

2. Sampling in the two-dimensional space

Basics on image sampling
The concept of spatial frequencies
Images of limited bandwidth
Two-dimensional sampling
Image reconstruction from its samples
The Nyquist rate. The alias effect and spectral replicas superposition
The sampling theorem in the two-dimensional case
Non-rectangular sampling grids and interlaced sampling
The optimal sampling
Practical limitations in sampling and reconstruction

3. Image quantization
4. The optimal quantizer
The uniform quantizer
5. Visual quantization
Contrast quantization
Pseudo-random noise quantization
Halftone image generation
Color image quantization

1. Introduction

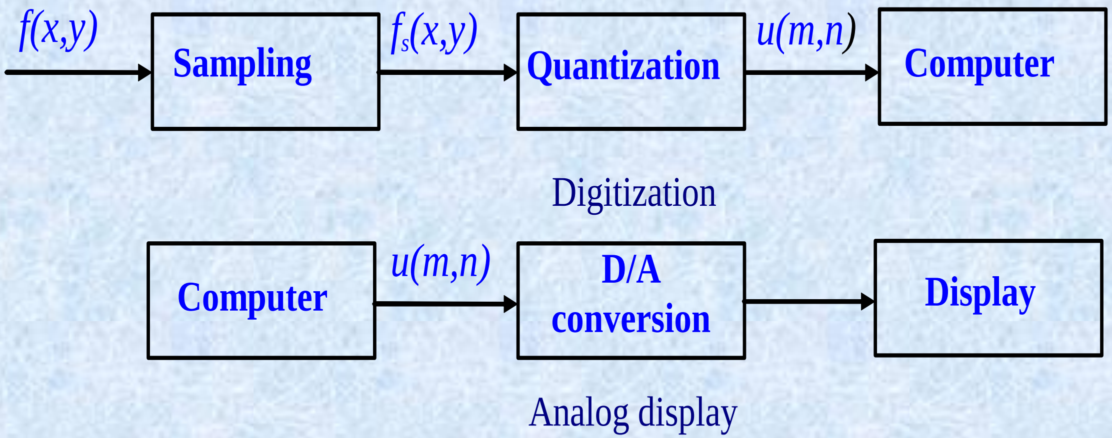
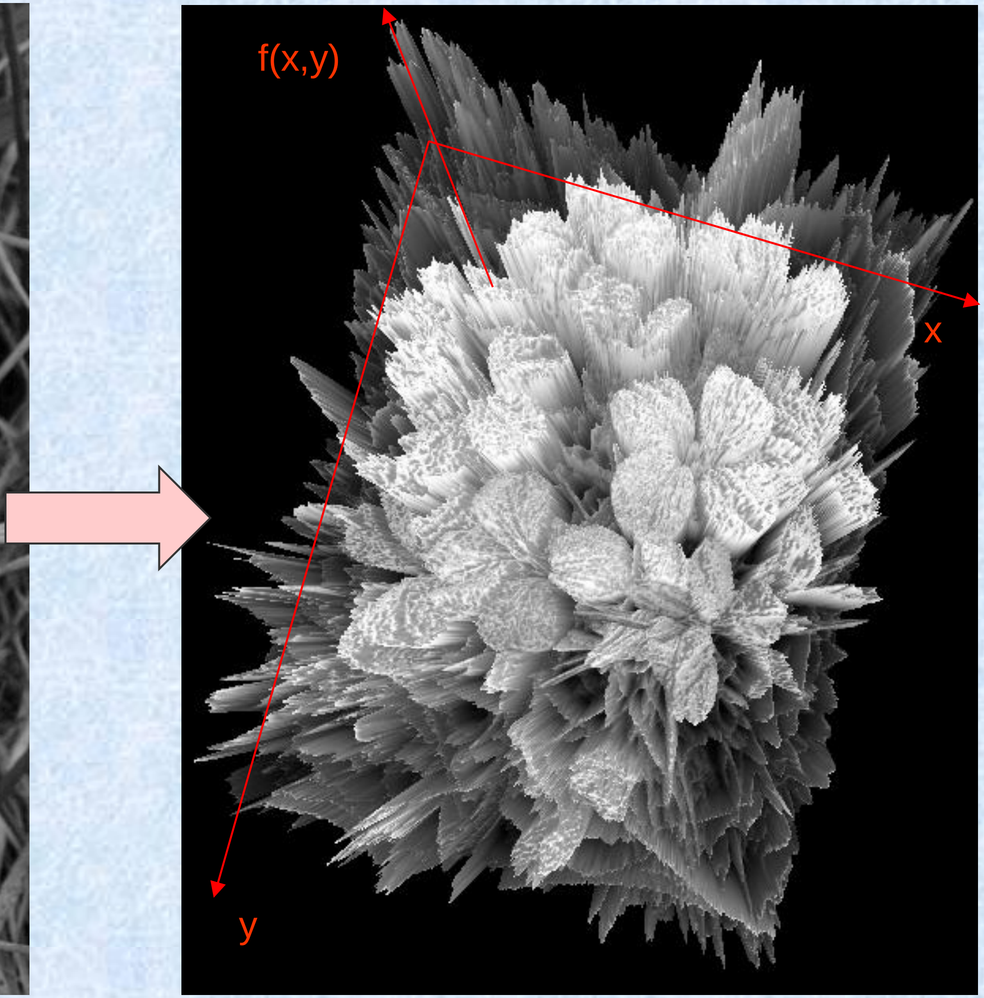


Fig 1 Image sampling and quantization / Analog image display

2. Sampling in the two-dimensional space

Basics on image sampling



The concept of spatial frequencies

- ***Grey scale images*** *can be seen as a 2-D generalization of time-varying signals* (both in the ***analog*** and in the ***digital*** case); the following equivalence applies:

1-D signal (time varying)	***2-D signal (grey scale image)***
Time coordinate t	Space coordinates x,y
Instantaneous value: f(t)	Brightness level, point-wise: f(x,y)
A 1-D signal that doesn't vary in time (is constant) = has 0 A.C. component, and only a D.C. component	A perfectly uniform image (it has the same brightness in all spatial locations); the D.C. component = the brightness in any point
The frequency content of a 1-D signal is proportional to the speed of variation of its instantaneous value in time: $\nu_{max} \sim \max(df/dt)$	The frequency content of an image (2-D signal) is proportional to the speed of variation of its instantaneous value in **space:** $\nu_{max,x} \sim \max(df/dx)$; $\nu_{max,y} \sim \max(df/dy)$ => $\nu_{max,x}$, $\nu_{max,y}$ = **"spatial frequencies"**
Discrete 1-D signal: described by its samples => a vector: $\mathbf{u}=[u(0)\ u(1)\ \ldots\ u(N\text{-}1)]$, N samples; *the position of the sample = the discrete time moment*	Discrete image (2-D signal): described by its samples, but in 2-D => a **matrix:** $\mathbf{U}[M\times N]$, $\mathbf{U}=\{u(m,n)\}$, $m=0,1,\ldots,M\text{-}1$; $n=0,1,\ldots,N\text{-}1$.
The spectrum of the time varying signal = the real part of the Fourier transform of the signal, $F(\omega)$; $\omega=2\pi\nu$.	The spectrum of the image = real part of the Fourier transform of the image = 2-D generalization of 1-D Fourier transform, $F(\omega_x,\omega_y)$ $\omega_x=2\pi\nu_x$; $\omega_y=2\pi\nu_y$

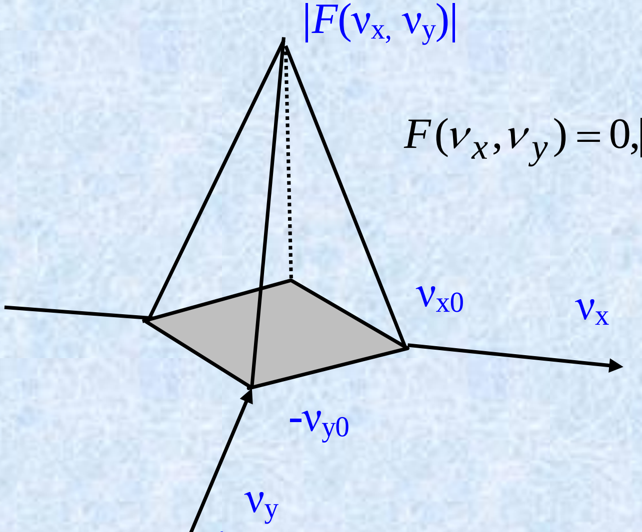
Images of limited bandwidth

- ***Limited bandwidth image = 2-D signal with finite spectral support:***

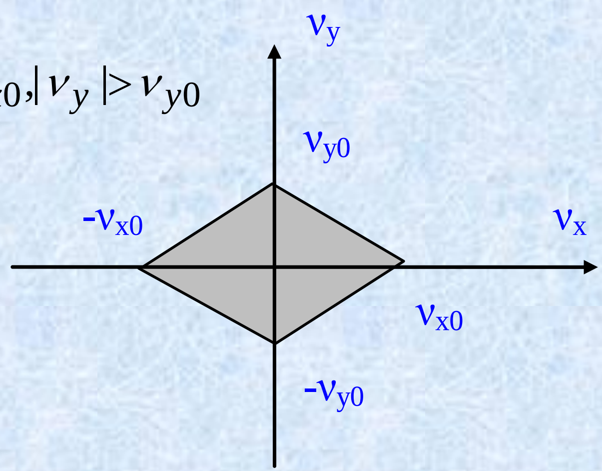
$F(\nu_x, \nu_y)$ = the Fourier transform of the image:

$$F(\nu_x,\nu_y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j\cdot\omega_x\cdot x}e^{-j\cdot\omega_y\cdot y}dxdy=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j\cdot 2\pi\cdot(\nu_x\cdot x+\nu_y\cdot y)}dxdy.$$

$$F(\nu_x,\nu_y)=0, |\nu_x|>\nu_{x0}, |\nu_y|>\nu_{y0}$$



The Fourier transform of the limited spectrum image

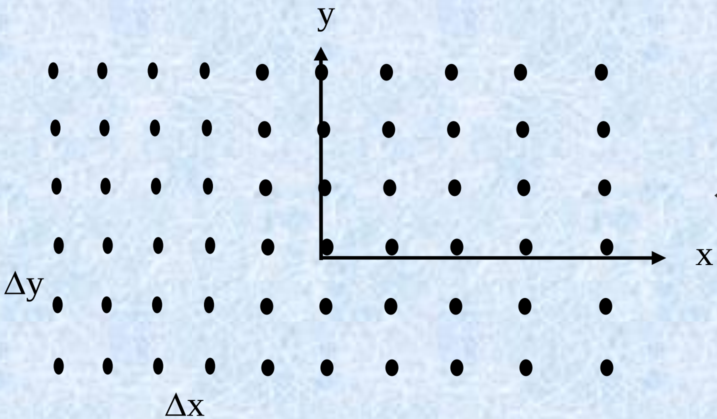


The spectral support region

The spectrum of a limited bandwidth image and its spectral support

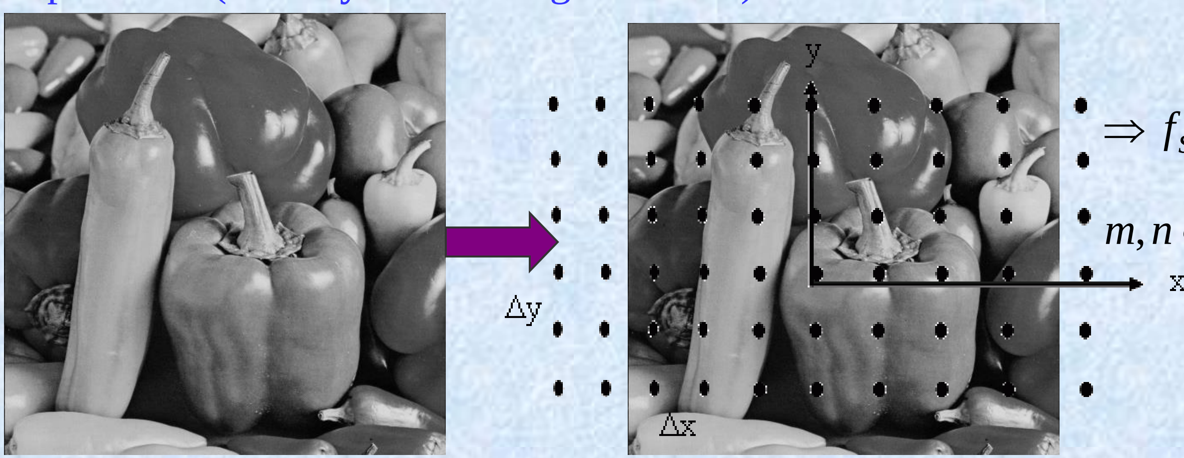
Two-dimensional sampling (1)

The common sampling grid = the uniformly spaced, rectangular grid:



$$g_{(\Delta x,\Delta y)}(x,y)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\delta(x-m\Delta x,y-n\Delta y)$$

Image sampling = read from the original, spatially continuous, brightness function f(*x*,*y*), **only** in the black dots positions (⇔ **only** where the grid allows):



$$\Rightarrow f_S(x,y)=\begin{cases} f(x,y), & x=m\Delta x, y=n\Delta y \\ 0, & otherwise \end{cases},$$

$m,n\in\mathbf{Z}.$

$$\Leftrightarrow f_S(x,y)=f(x,y)g_{(\Delta x,\Delta y)}(x,y)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}f(m\Delta x,n\Delta y)\delta(x-m\Delta x,y-n\Delta y)$$

Two-dimensional sampling (2)

• **Question:** How to choose the values Δx, Δy to achieve:

-the representation of the digital image by **the min. number of samples,**

-at (ideally) **no loss of information?**

(I. e.: for a perfectly uniform image, only 1 sample is enough to completely represent the image => sampling can be done with very large steps; on the opposite – if the brightness varies very sharply => very many samples needed)

⇒**The sampling intervals** Δx, Δy needed to have no loss of information **depend on the spatial frequency content of the image.**

⇒Sampling conditions for no information loss – derived by examining the **spectrum of the image** ⇔ by performing the Fourier analysis:

$$f_S(x,y) = f(x,y)g_{(\Delta x,\Delta y)}(x,y) \Rightarrow \text{Fourier transform:}$$

$$F_S(\nu_x,\nu_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_S(x,y)\cdot e^{-j2\pi\cdot\nu_x\cdot x}\cdot e^{-j2\pi\cdot\nu_y\cdot y}dxdy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\cdot g_{(\Delta x,\Delta y)}(x,y)\cdot e^{-j2\pi\cdot\nu_x\cdot x}\cdot e^{-j2\pi\cdot\nu_y\cdot y}dxdy.$$

⇒The ***sampling grid*** function $g_{(\Delta x,\Delta y)}$ is periodical with period $(\Delta x, \Delta y)$ => can be expressed by its Fourier series expansion:

$$g_{(\Delta x,\Delta y)}(x,y) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} a(k,l)\cdot e^{j2\pi\cdot\frac{k}{\Delta x}\cdot x}\cdot e^{j2\pi\cdot\frac{l}{\Delta y}\cdot y},$$

where:

$$a(k,l) = \frac{1}{\Delta x}\cdot\frac{1}{\Delta y}\cdot\int_0^{\Delta x}\int_0^{\Delta y} g_{(\Delta x,\Delta y)}(x,y)e^{-j2\pi\cdot\frac{k}{\Delta x}\cdot x}\cdot e^{-j2\pi\cdot\frac{l}{\Delta y}\cdot y}dxdy.$$

Two-dimensional sampling (3)

Since:

$$\text{for } (x,y)\in[0;\Delta x)\times[0;\Delta y),\quad g_{(\Delta x,\Delta y)}(x,y)=\begin{cases}1, & if\ \ x=y=0\\ 0, & otherwise\end{cases},$$

$$\Rightarrow a(k,l)=\frac{1}{\Delta x}\cdot\frac{1}{\Delta y},\quad \forall k,l\in(-\infty;\infty).$$

Therefore the Fourier transform of f_S is:

$$F_S(\nu_x,\nu_y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\cdot\left(\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\frac{1}{\Delta x\cdot\Delta y}\cdot e^{j\frac{2\pi\cdot k\cdot x}{\Delta x}}\cdot e^{j\frac{2\pi\cdot l\cdot y}{\Delta y}}\right)\cdot e^{-j2\pi\cdot\nu_x\cdot x}\cdot e^{-j2\pi\cdot\nu_y\cdot y}dxdy$$

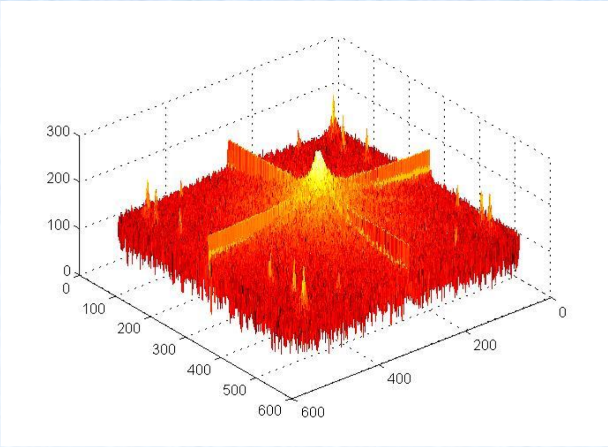
$$=> F_S(\nu_x,\nu_y)=\frac{1}{\Delta x\cdot\Delta y}\cdot\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} f(x,y)\cdot e^{-j2\pi\cdot x\cdot\left(\nu_x-\frac{k}{\Delta x}\right)}\cdot e^{-j2\pi\cdot y\cdot\left(\nu_y-\frac{l}{\Delta y}\right)}\right)dxdy$$

$$\Leftrightarrow F_S(\nu_x,\nu_y)=\frac{1}{\Delta x\cdot\Delta y}\cdot\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\cdot e^{-j2\pi\cdot x\cdot\left(\nu_x-\frac{k}{\Delta x}\right)}\cdot e^{-j2\pi\cdot y\cdot\left(\nu_y-\frac{l}{\Delta y}\right)}dxdy$$

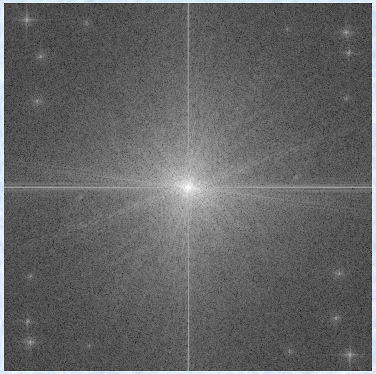
$$\Leftrightarrow F_S(\nu_x,\nu_y)=\frac{1}{\Delta x\cdot\Delta y}\cdot\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} F\left(\nu_x-\frac{k}{\Delta x},\nu_y-\frac{l}{\Delta y}\right).$$

⇔ ***The spectrum of the sampled image = the collection of an infinite number of scaled spectral replicas of the spectrum of the original image,*** centered at multiples of spatial frequencies 1/Δx, 1/ Δy.

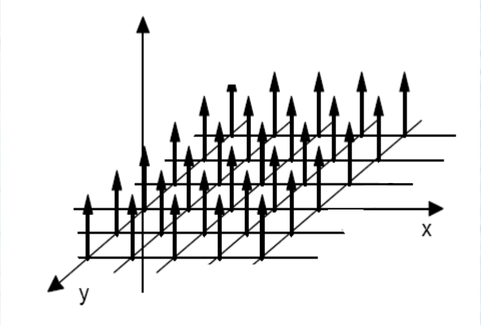
Original image



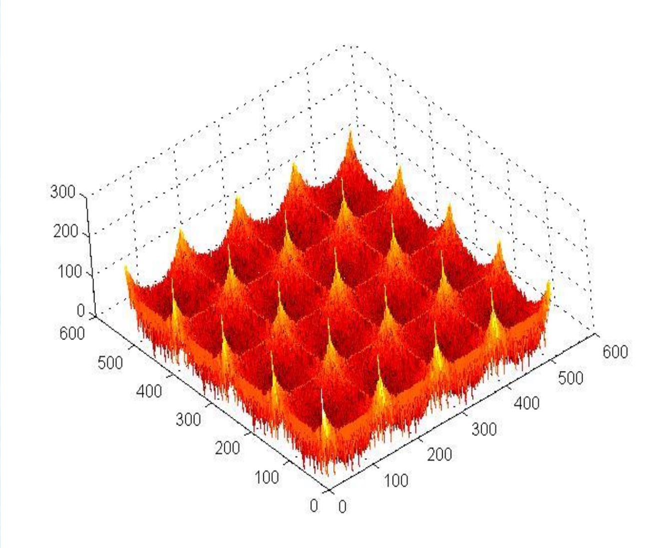
Original image spectrum – 3D



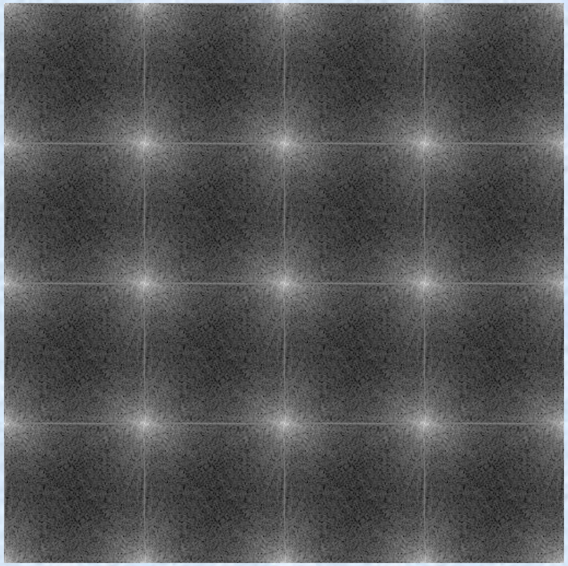
Original image spectrum – 2D



2-D rectangular sampling grid



Sampled image spectrum – 3D



Sampled image spectrum – 2D

Image reconstruction from its samples

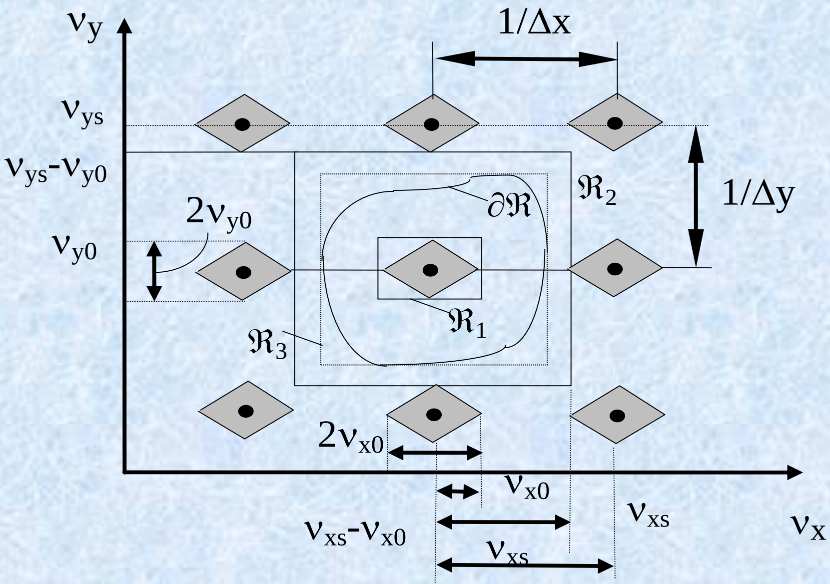


Fig.4 The sampled image spectrum

$$\nu_{xs} > 2\nu_{x0}, \quad \nu_{ys} > 2\nu_{y0} \qquad \Delta x < \frac{1}{2\nu_{x0}}, \Delta y < \frac{1}{2\nu_{y0}}$$

$$H(\nu_x,\nu_y) = \begin{cases} \frac{1}{(\nu_{xs}\nu_{ys})}, (\nu_x,\nu_y) \in \Re \\ 0, \qquad \text{otherwise} \end{cases}$$

$$\tilde{F}(\nu_x,\nu_y) = H(\nu_x,\nu_y)F_s(\nu_x,\nu_y) = F(\nu_x,\nu_y) \Leftrightarrow$$

$$\tilde{f}(x,y) = h(x,y) * f_s(x,y)$$

$$\Leftrightarrow \tilde{f}(x,y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f_s(m\Delta x, n\Delta y) h(x - m\Delta x, y - n\Delta y)$$

Let us assume that the filtering region R is rectangular, at the middle distance between two spectral replicas:

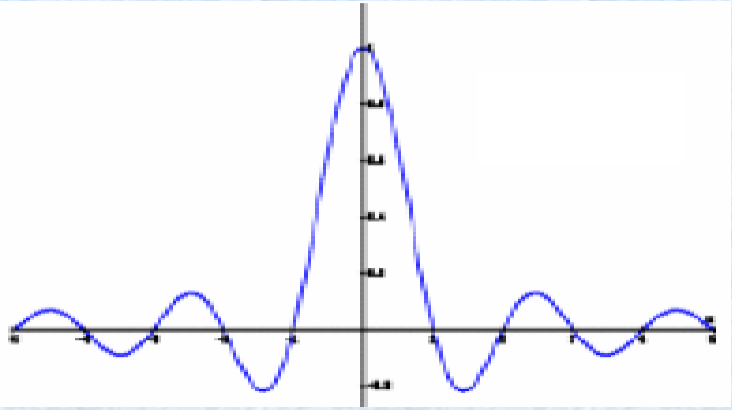
$$H(\nu_x,\nu_y) = \begin{cases} \frac{1}{(\nu_{xs}\nu_{ys})}, |\nu_x| < \frac{\nu_{xs}}{2} \text{ and } |\nu_y| < \frac{\nu_{ys}}{2} \\ 0, \qquad \text{otherwise} \end{cases} \Rightarrow h(x,y) = \frac{\sin(\pi x \nu_{xs})}{\pi x \nu_{xs}} \cdot \frac{\sin(\pi y \nu_{ys})}{\pi y \nu_{ys}}$$

$$\Rightarrow \tilde{f}(x,y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f_s(m\Delta x, n\Delta y) h(x - m\Delta x, y - n\Delta y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} f_s(m\Delta x, n\Delta y)\frac{\sin(\pi(x\nu_{xs} - m))}{\pi(x\nu_{xs} - m)} \cdot \frac{\sin(\pi(y\nu_{ys} - n))}{\pi(y\nu_{ys} - n)}$$

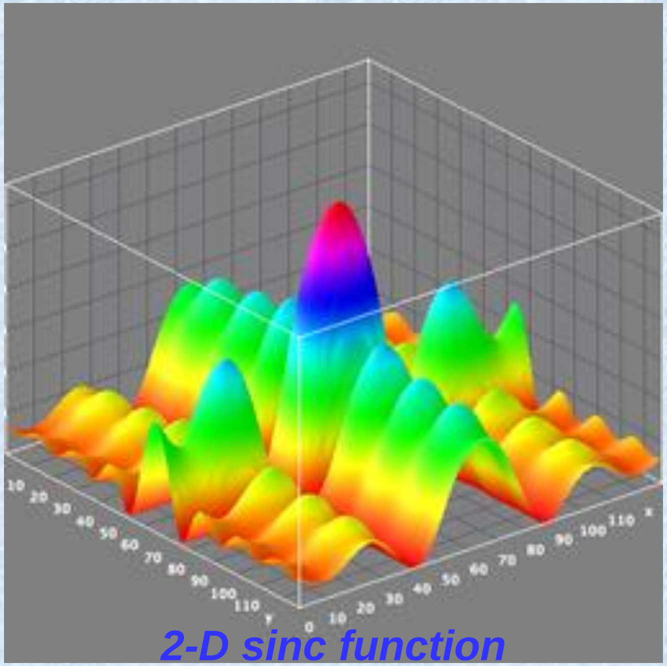
$$\Leftrightarrow \tilde{f}(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f_S(m\Delta x, n\Delta y)\,\text{sinc}(x\nu_{xS} - m)\,\text{sinc}(y\nu_{yS} - n), \quad \text{where } \text{sinc}(a) = \frac{\sin \pi a}{\pi a}$$

Since the sinc function has infinite extent => it is impossible to implement in practice the ideal LPF $\Rightarrow$ it is impossible to reconstruct in practice an image from its samples without error if we sample it at the Nyquist rates.

Practical solution: sample the image at higher spatial frequencies + implement a real LPF (as close to the ideal as possible).



1-D sinc function



2-D sinc function

The Nyquist rate. The aliasing. The fold-over frequencies

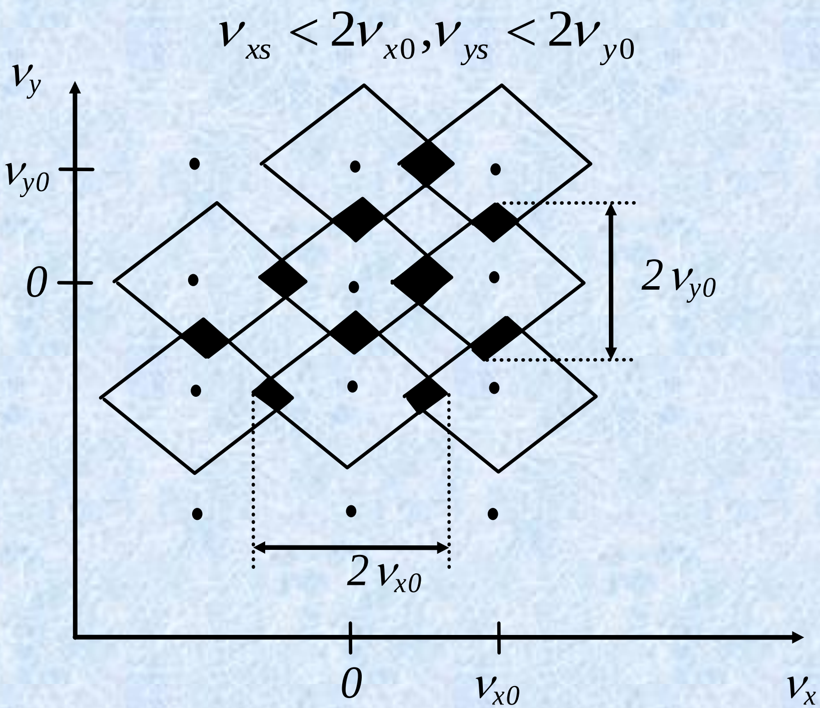
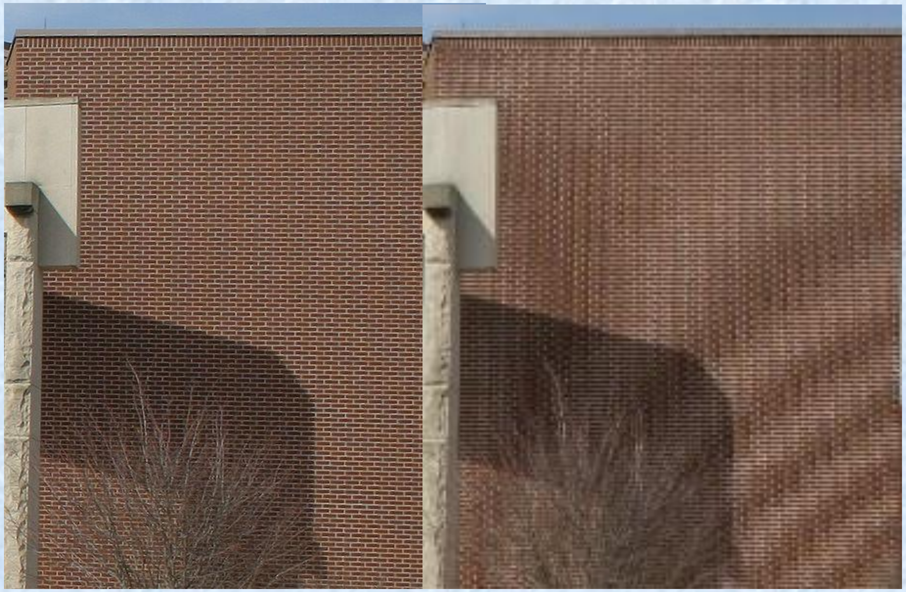
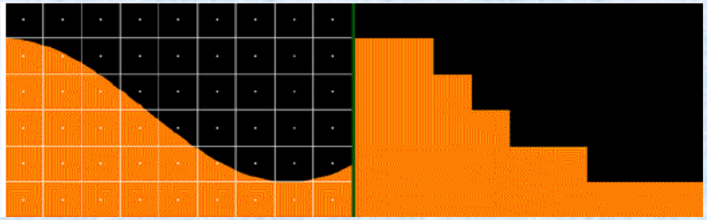


Fig. 5 Aliasing – fold-over frequencies



The Moire effect



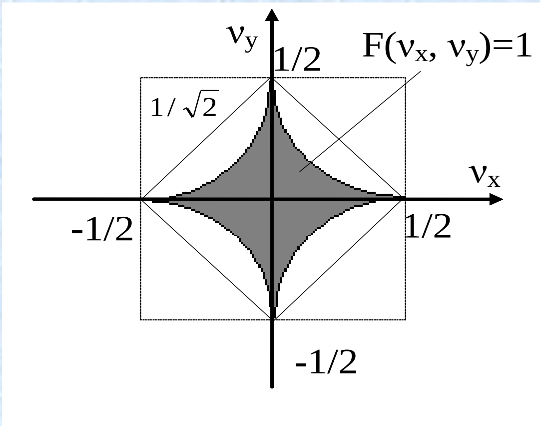
"Jagged" boundaries

Note: Aliasing may also appear in the reconstruction process, due to the imperfections of the filter!

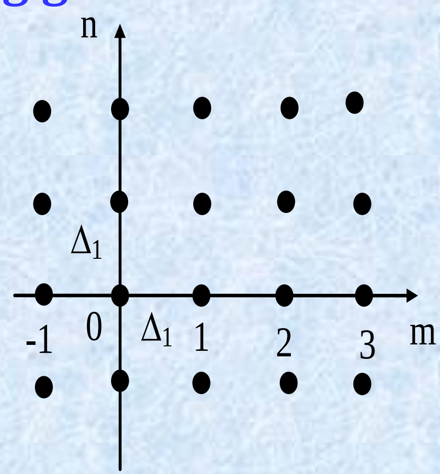
How to avoid aliasing if cannot increase the sampling frequencies?

By a LPF on the image applied prior to sampling!

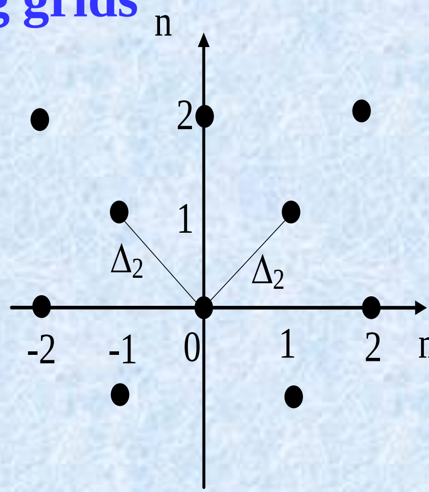
Non-rectangular sampling grids. Interlaced sampling grids



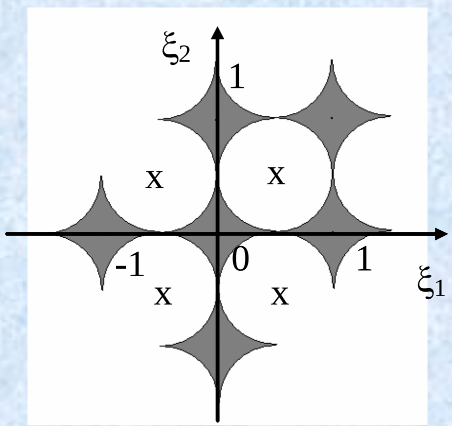
a) Image spectrum



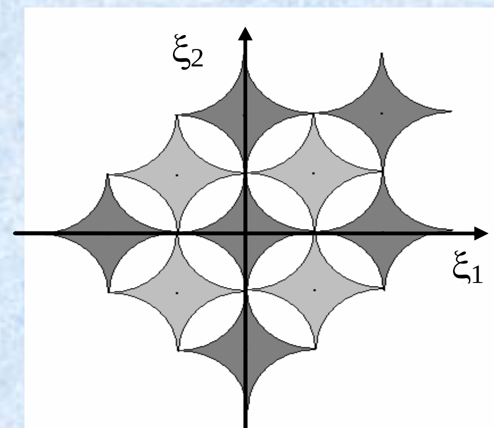
b) Rectangular grid G_1



c) Interlaced grid G_2



d) The spectrum using G_1

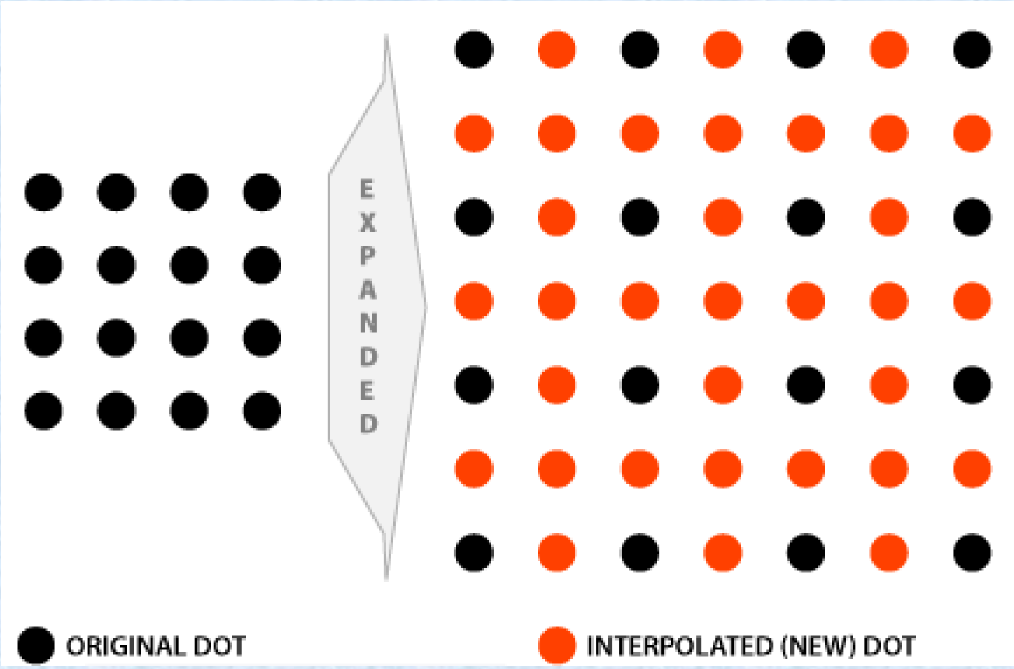


e) The spectrum using G_2

Interlaced sampling

Optimal sampling = Karhunen-Loeve expansion: $f(x,y) = \sum \sum_{m,n=0}^{\infty} a_{m,n} \Phi_{m,n}$

Image reconstruction from its samples in the real case



The question is: what to fill in the "interpolated" (new) dots?
Several interpolation methods are available; ideally – sinc function in the spatial domain; in practice – simpler interpolation methods (i.e. approximations of LPFs).

Image interpolation filters:

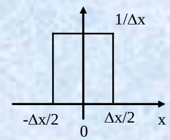
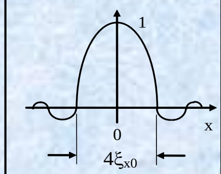
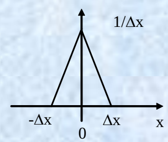
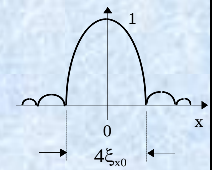
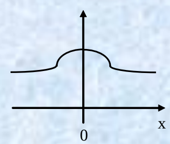
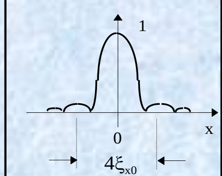
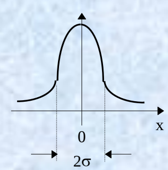
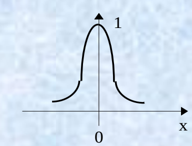
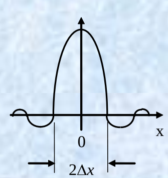
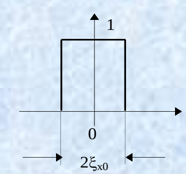
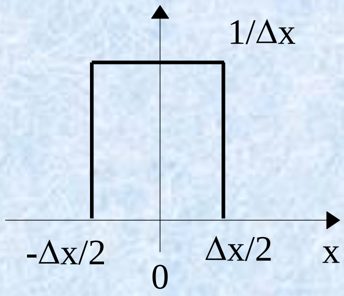
The 1-D interpolation function	Graphical representation	p(x)	The 2-D interpolation function $p_a(x,y)=p(x)p(y)$	Frequency response $p_a(\xi_1,\xi_2)$	$p_a(\xi_1,0)$
Rectangular (zero-order filter) $p_0(x)$		$\frac{1}{\Delta x} rect\left(\frac{x}{\Delta x}\right)$	$p_0(x)p_0(y)$	$\text{sinc}\left(\frac{\xi_1}{2\xi_{x0}}\right)\text{sinc}\left(\frac{\xi_2}{2\xi_{y0}}\right)$	
Triangular (first order filter) $p_1(x)$		$\frac{1}{\Delta x} tri\left(\frac{x}{\Delta x}\right)$ $p_0(x)\otimes p_0(x)$	$p_1(x)p_1(y)$	$\left[\text{sinc}\left(\frac{\xi_1}{2\xi_{x0}}\right)\text{sinc}\left(\frac{\xi_2}{2\xi_{y0}}\right)\right]^2$	
n-order filter *n*=2, quadratic *n*=3, cubic spline $p_n(x)$		$p_0(x)\otimes\cdots\otimes p_0(x)$ n convolu@i	$p_n(x)p_n(y)$	$\left[\text{sinc}\left(\frac{\xi_1}{2\xi_{x0}}\right)\text{sinc}\left(\frac{\xi_2}{2\xi_{y0}}\right)\right]^{n+1}$	
Gaussian $p_g(x)$		$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{x^2}{2\sigma^2}\right]$	$\frac{1}{2\pi\sigma^2}\exp\left[-\frac{(x^2+y^2)}{2\sigma^2}\right]$	$\exp\left[-2\pi^2\sigma^2(\xi_1^2+\xi_2^2)\right]$	
Sinc		$\frac{1}{\Delta x}\text{sinc}\left(\frac{x}{\Delta x}\right)$	$\frac{1}{\Delta x\Delta y}\text{sinc}\left(\frac{x}{\Delta x}\right)\text{sinc}\left(\frac{y}{\Delta y}\right)$	$rect\left(\frac{\xi_1}{2\xi_{x0}}\right)rect\left(\frac{\xi_2}{2\xi_{y0}}\right)$	

Image interpolation examples:

1. Rectangular (zero-order) filter, or *nearest neighbour filter*, or *box filter*:



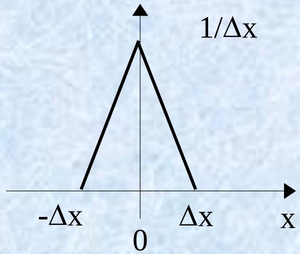
Original

Sampled

Reconstructed

Image interpolation examples:

2. Triangular (first-order) filter, or *bilinear filter*, or *tent filter*:



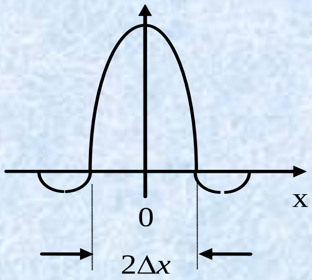
Original

Sampled

Reconstructed

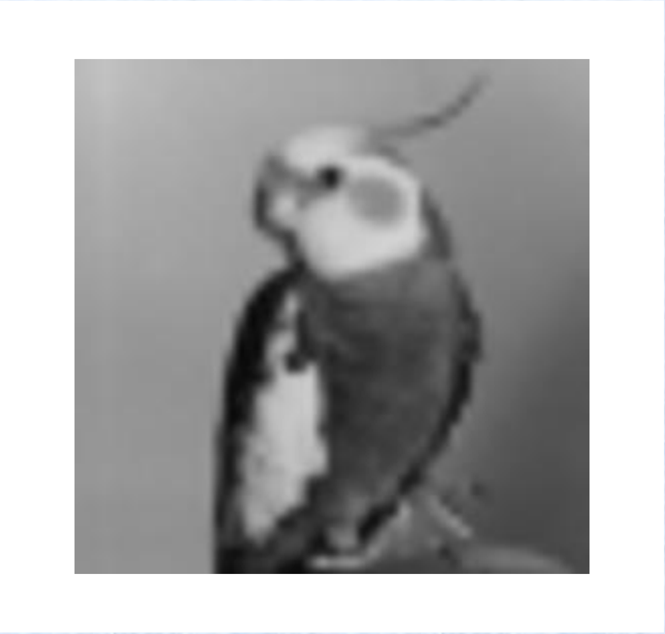
Image interpolation examples:

3. Cubic interpolation filter, or *bicubic filter* – begins to better approximate the sinc function:



Original

Sampled



Reconstructed

Practical limitations in image sampling and reconstruction

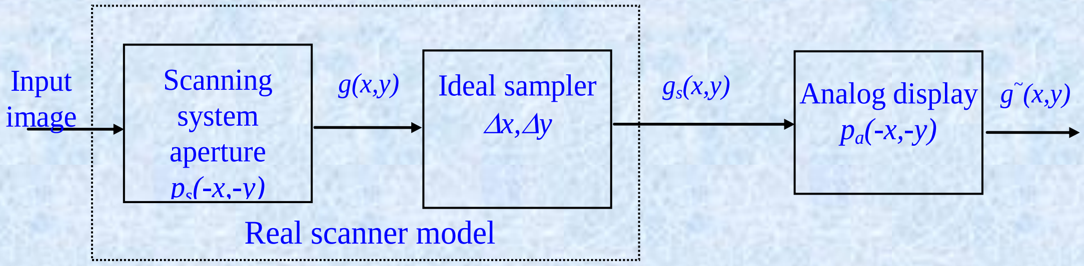


Fig. 7 The block diagram of a real sampler & reconstruction (display) system

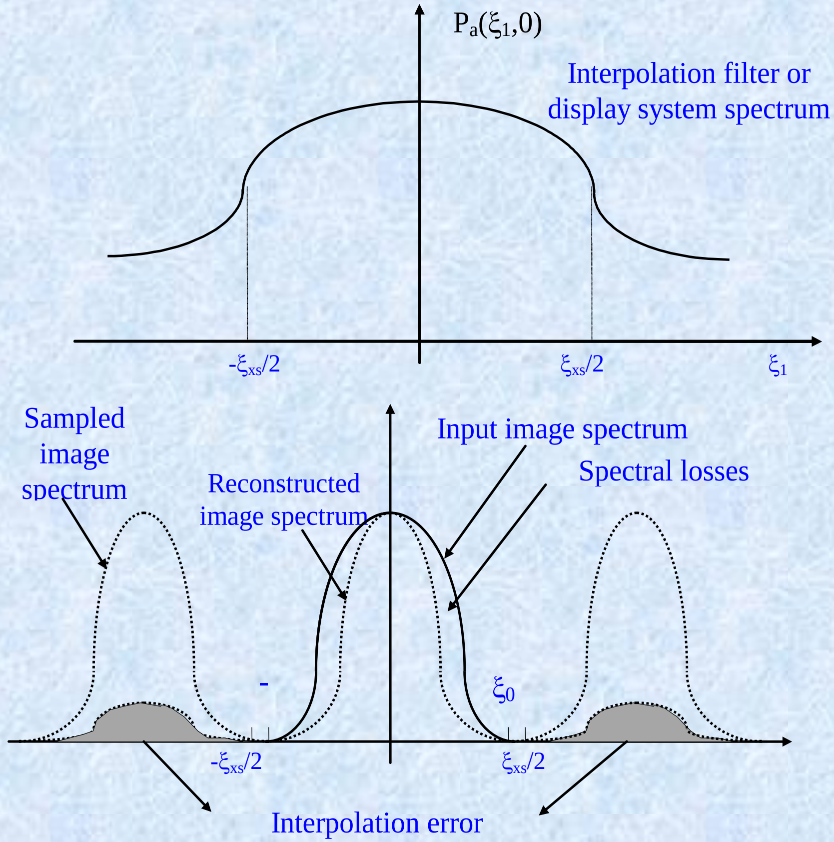


Fig. 8 The real effect of the interpolation

3. Image quantization

3.1. Overview

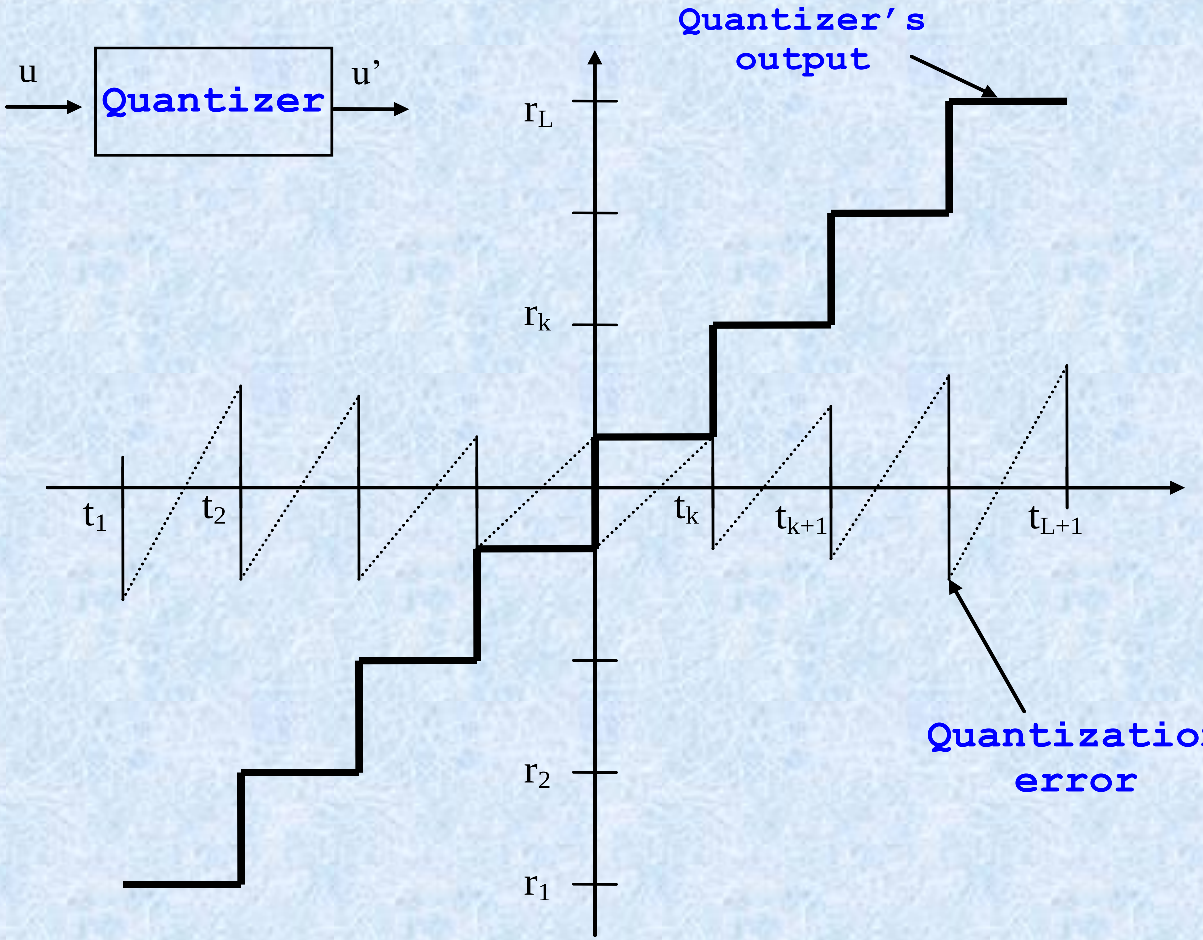


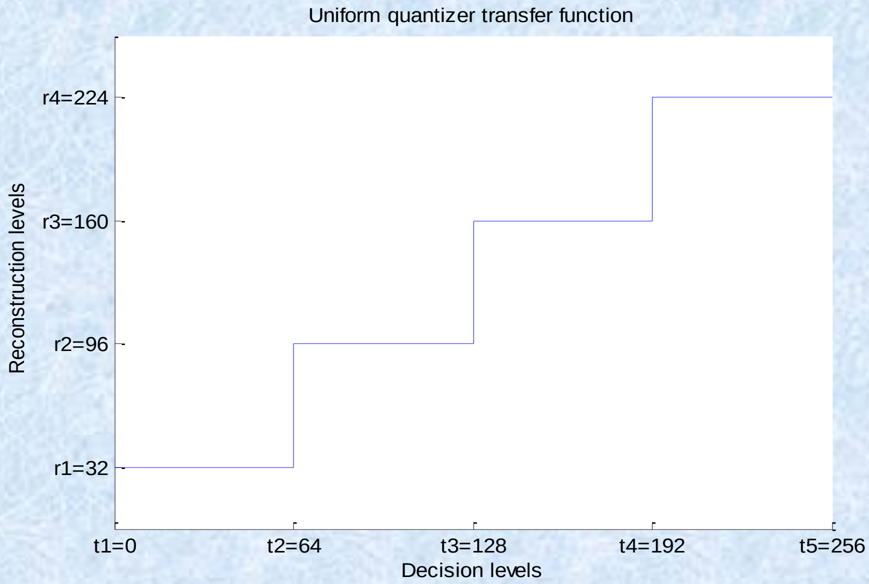
Fig. 9 The quantizer's transfer function

3.2. The uniform quantizer

The quantizer's design:

- *Denote the input brightness range:* $u \in [l_{min}; L_{Max}]$
- *Let* B *– the number of bits of the quantizer => L=2^B reconstruction levels*
- *The expressions of the decision levels:*

E.g. B=2 => L=4



$$t_1 = l_{\min}; \quad t_{L+1} = L_{Max}$$

$$t_k - t_{k-1} = t_{k+1} - t_k = \text{constant} = q$$

$$q = \frac{t_{L+1} - t_1}{L}, t_k = t_{k-1} + q \Rightarrow q = \frac{L_{Max} - l_{min}}{L}$$

- *The expressions of the reconstruction levels:*

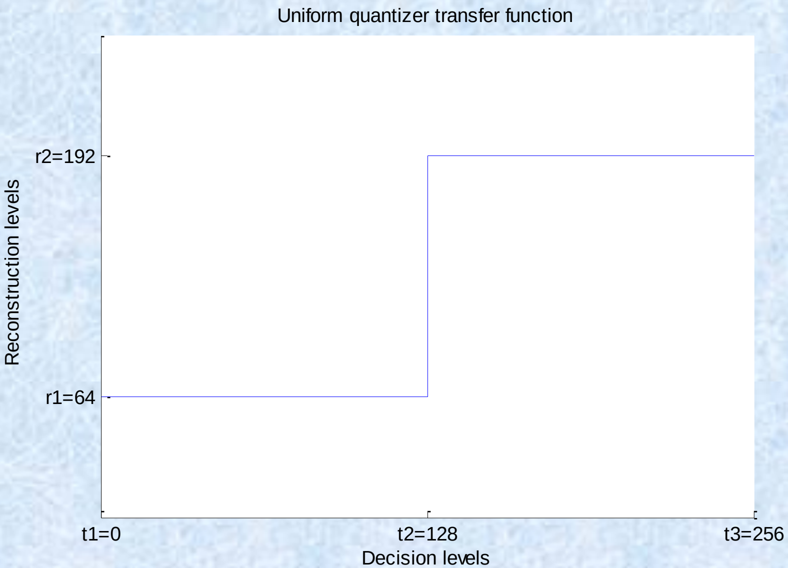
$$r_k = \frac{t_k + t_{k+1}}{2} \Rightarrow r_k = t_k + \frac{q}{2}$$

- *Computation of the quantization error: for a given image of size M×N pixels,* **U** *– non-quantized, and* **U'** *– quantized => we estimate the MSE:*

$$\varepsilon = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left(u(m,n) - u'(m,n)\right)^2 = \sum_{k=1}^{L}\int_{t_k}^{t_{k+1}} (u - r_k)^2 h_{lin,U}(u)du$$

Examples of uniform quantization and the resulting errors:

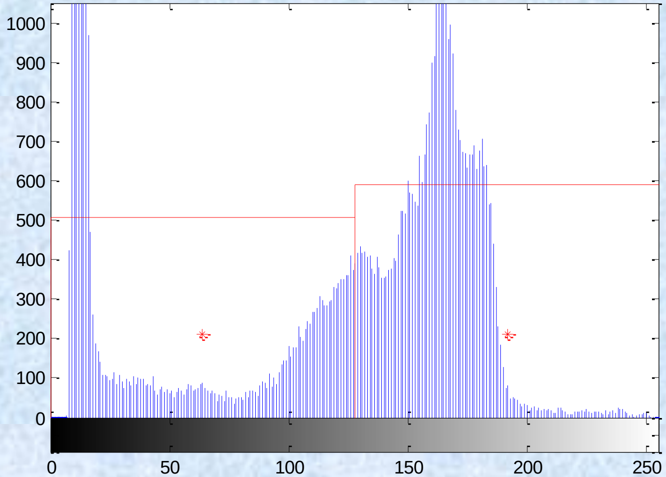
B=1 => L=2



Non-quantized image

Quantized image

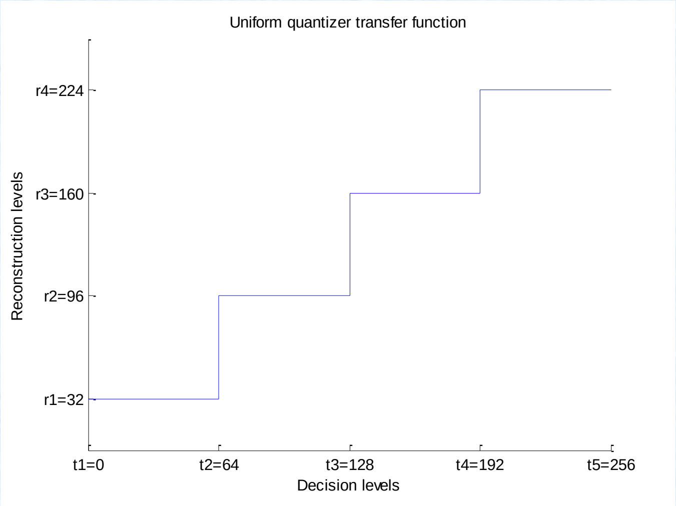
Quantization error; MSE=36.2



The histogram of the non-quantized image

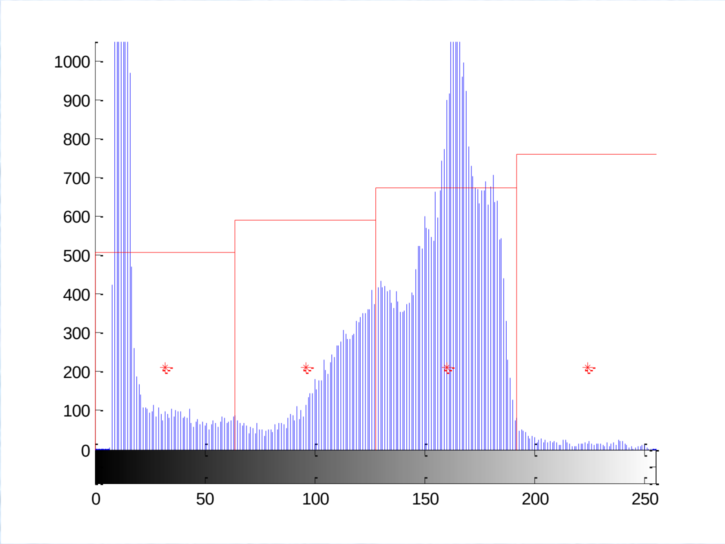
Examples of uniform quantization and the resulting errors:

B=2 => L=4



Non-quantized image

Quantized image

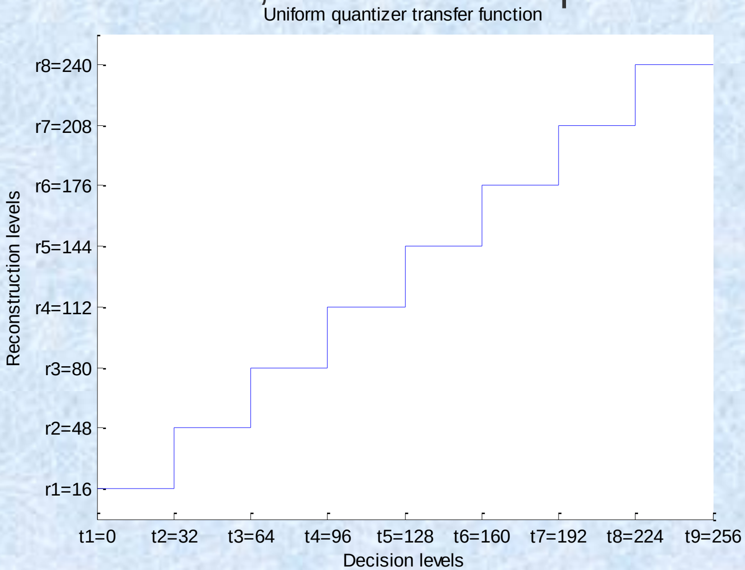


The histogram of the non-quantized image

Quantization error; MSE=15

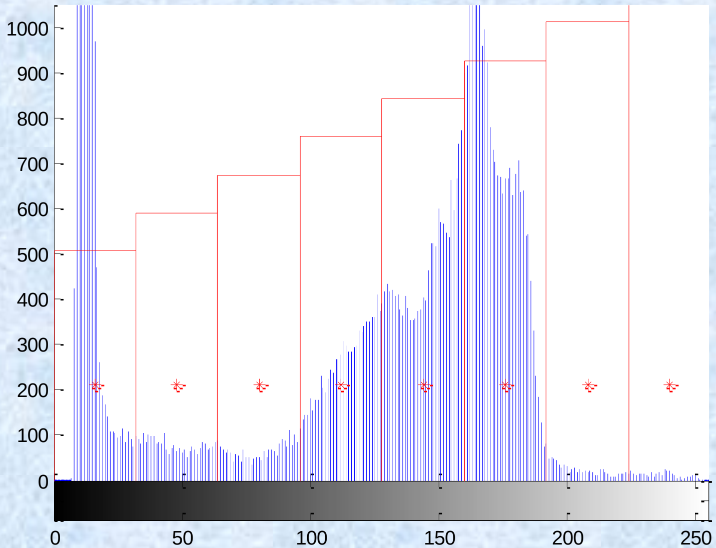
Examples of uniform quantization and the resulting errors:

B=3 => L=8; false contours present



Non-quantized image

Quantized image



The histogram of the non-quantized image

Quantization error; MSE=7.33

3.2. The optimal (MSE) quantizer (the Lloyd-Max quantizer)

$$e = E[(u-u')^2] = \int_{t_1}^{t_{L+1}} (u-u')^2 h_u(u)du \quad \Rightarrow \quad \varepsilon = \sum_{i=1}^{L} \int_{t_i}^{t_{i+1}} (u-r_i)^2 h_u(u)du$$

$$\frac{\partial \varepsilon}{\partial t_k} = \left((t_k - r_{k-1})^2 - (t_k - r_k)^2\right) h_u(t_k) = 0$$

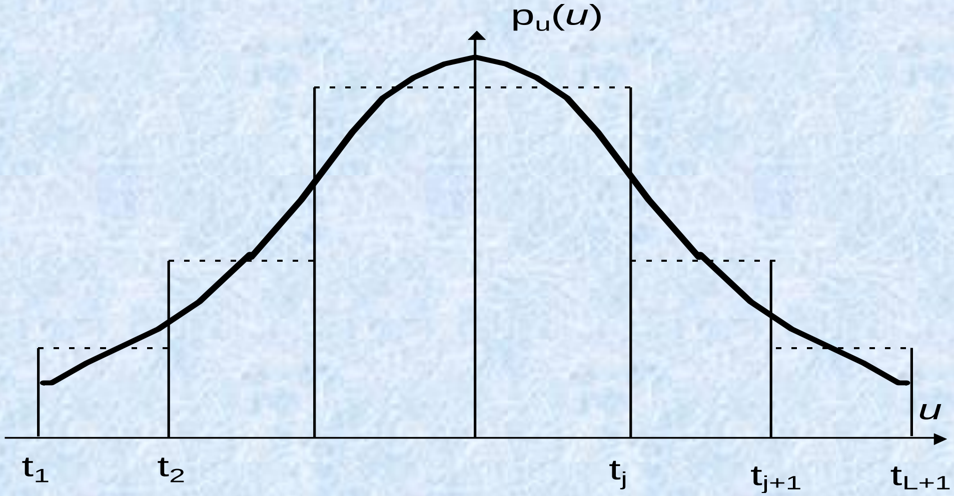
$$\frac{\partial \varepsilon}{\partial r_k} = -2 \int_{t_k}^{t_{k+1}} (u - r_k) h_u(u)du = 0 \quad 1 \le k \le L$$

$$t_k = \frac{r_k + r_{k-1}}{2} \qquad r_k = \frac{\int_{t_k}^{t_{k+1}} u h_u(u)du}{\int_{t_k}^{t_{k+1}} h_u(u)du} = E\left[u \mid u \in \vartheta_k\right]$$

$$p_u(u) \cong p_u(\hat{t}_j), \qquad \hat{t}_j = \frac{1}{2}(t_j + t_{j+1}), \qquad t_j \le u < t_{j+1}$$

$$t_{k+1} \cong \frac{A \int_{t_1}^{z_k + t_1} [h_u(u)]^{-1/3} du}{\int_{t_1}^{t_{L+1}} [h_u(u)]^{-1/3} du} + t_1$$

$$\varepsilon = \frac{1}{12L^2}\left\{\int_{t_1}^{t_{L+1}} [h_u(u)]^{1/3} du\right\}^3$$

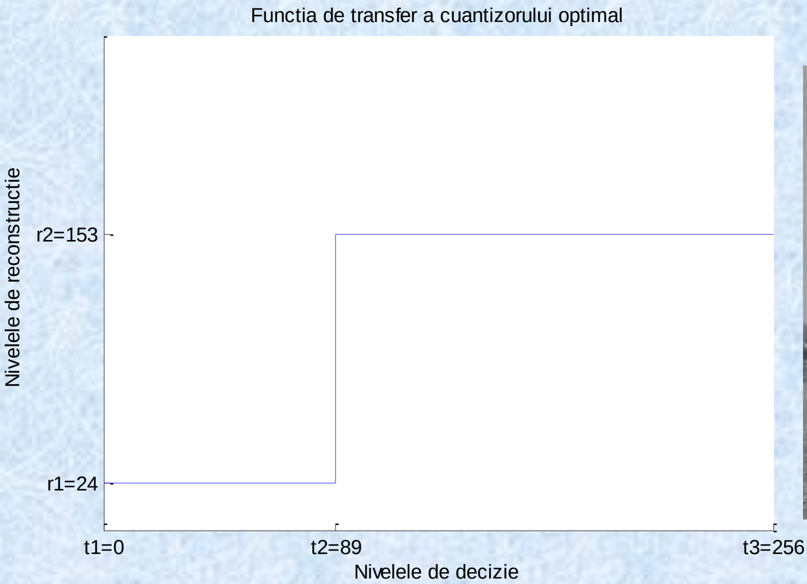


$$h_u(u) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(u-\mu)^2}{2\sigma^2}\right)$$ (Gaussian) , or $$h_u(u) = \frac{\alpha}{2}\exp\left(-\alpha|u-\mu|\right)$$ (Laplacian)

$\sigma^2 = \frac{2}{\alpha}$ (variance , μ- mean)

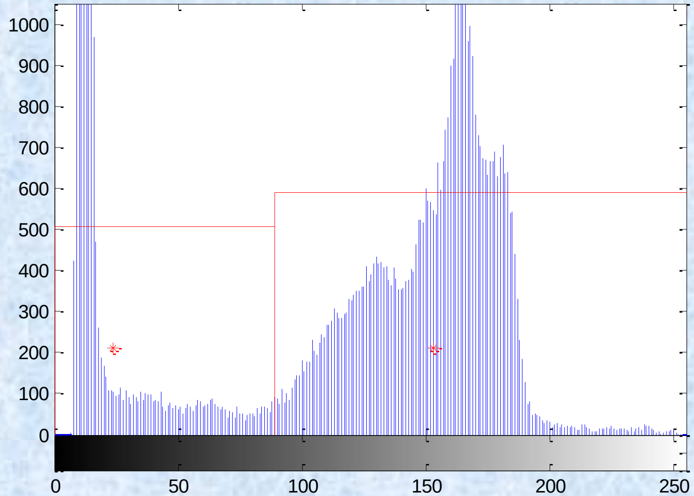
Examples of optimal quantization and the quantization error:

B=1 => L=2



Non-quantized image

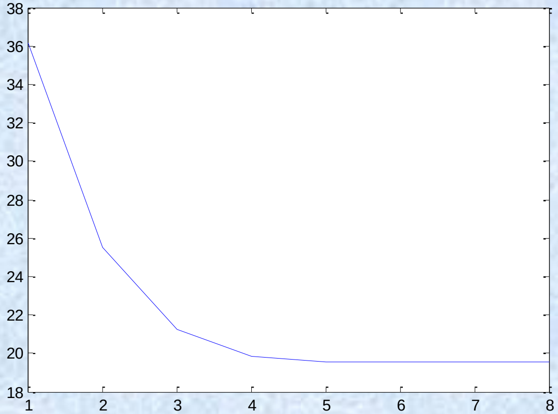
Quantized image



The non-quantized image histogram

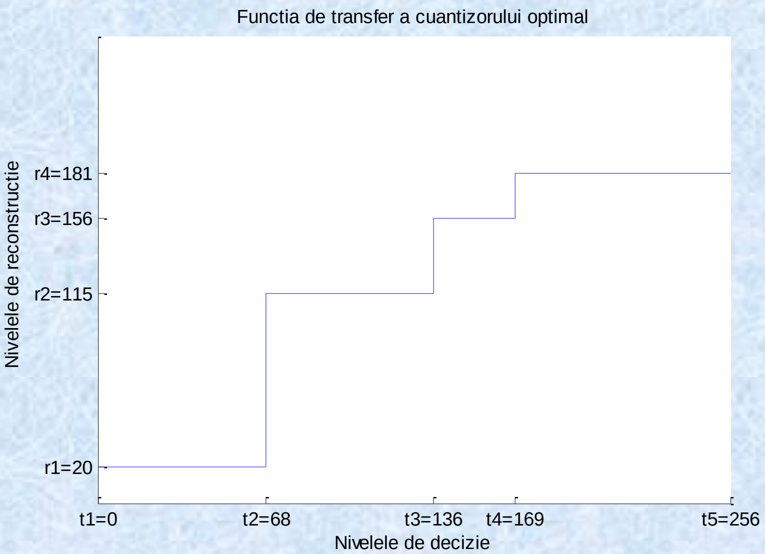
The quantization error; MSE=19.5

The evolution of MSE in the optimization, starting from the uniform quantizer



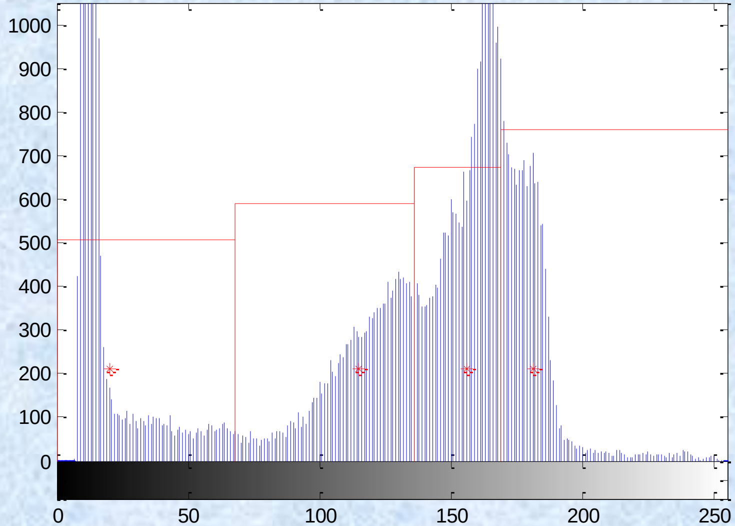
Examples of optimal quantization and the quantization error:

B=2 => L=4



Non-quantized image

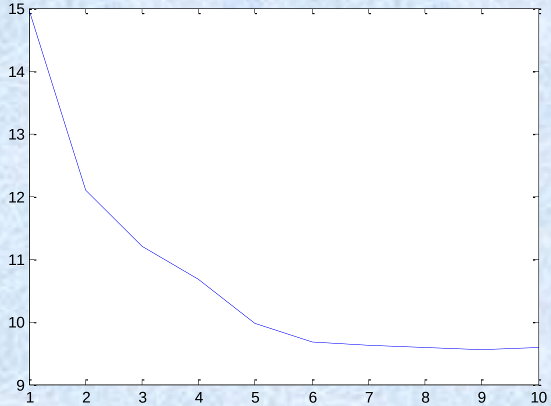
Quantized image



The non-quantized image histogram

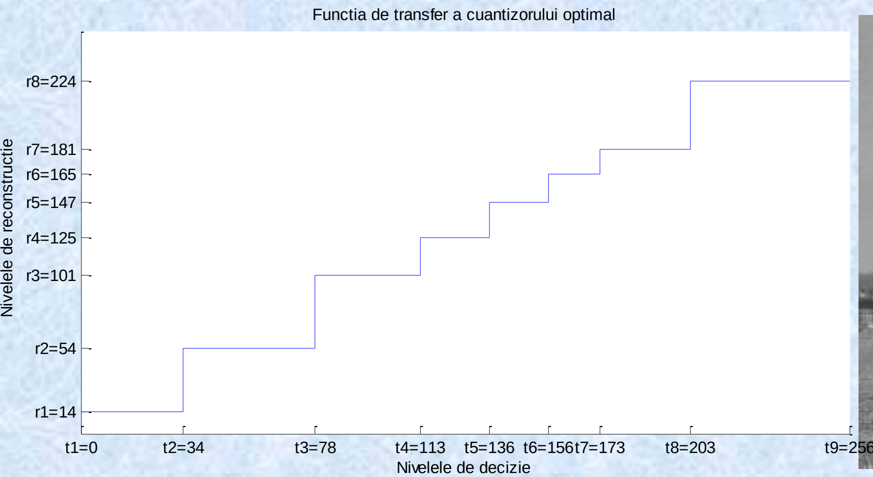
The quantization error; MSE=9.6

The evolution of MSE in the optimization, starting from the uniform quantizer



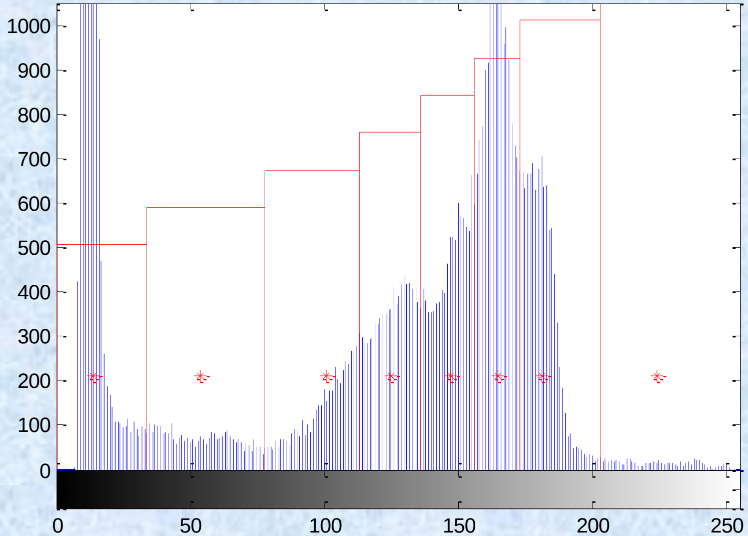
Examples of optimal quantization and the quantization error:

B=3 => L=8



Non-quantized image

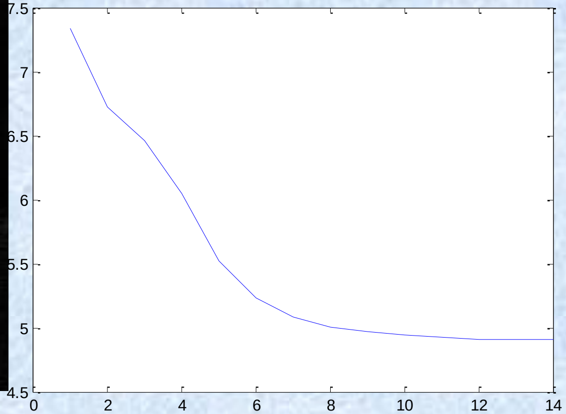
Quantized image



The non-quantized image histogram

The quantization error; MSE=5

The evolution of MSE in the optimization, starting from the uniform quantizer



3.3. The uniform quantizer = the optimal quantizer for the uniform grey level distribution:

$$h_u(u) = \begin{cases} \dfrac{1}{t_{L+1} - t_1}, & t_1 \le u \le t_{L+1} \\ 0 & otherwise \end{cases}$$

$$r_k = \frac{(t_{k+1}^2 - t_k^2)}{2(t_{k+1} - t_k)} = \frac{t_{k+1} + t_k}{2}$$

$$t_k = \frac{t_{k+1} + t_{k-1}}{2} \qquad t_k - t_{k-1} = t_{k+1} - t_k = \text{constant} = q$$

$$q = \frac{t_{L+1} - t_1}{L}, \qquad t_k = t_{k-1} + q, \qquad r_k = t_k + \frac{q}{2}$$

$$\varepsilon = \frac{1}{q} \int_{-q/2}^{q/2} u^2 du = \frac{q^2}{12}$$

$$\frac{\varepsilon}{\sigma_u^2} = 2^{-2B}, \text{therefore } SNR = 10log_{10} 2^{2B} = 6 \cdot B \ \ dB$$

3.4. Visual quantization methods

- In general – if B<6 (uniform quantization) or B<5 (optimal quantization) => the "contouring" effect (i.e. false contours) appears in the quantized image.
- The false contours ("contouring") = groups of neighbor pixels quantized to the same value <=> regions of constant gray levels; the boundaries of these regions are the false contours.
- The false contours do not contribute significantly to the MSE, but are very disturbing for the human eye => *it is important to reduce the visibility of the quantization error, not only the MSQE.*

⇒*Solutions:* visual quantization schemes, to hold quantization error below the level of visibility.

⇒Two main schemes: (a) contrast quantization; (b) pseudo-random noise quantization

Uniform quantization, B=4

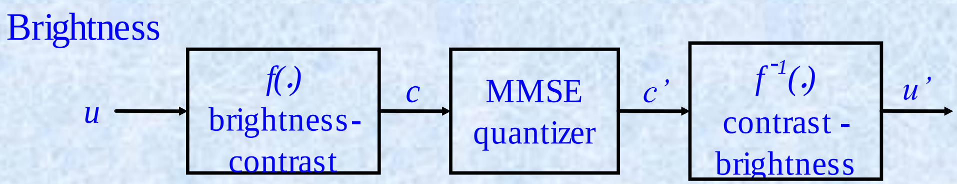
Optimal quantization, B=4

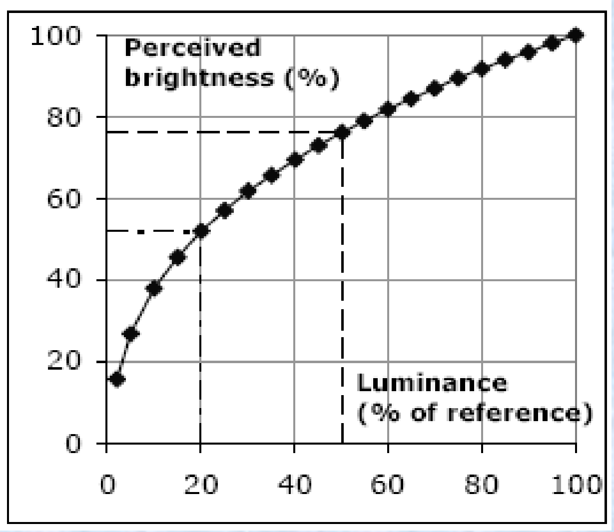
Uniform quantization, B=6

3.4. Visual quantization methods

a. Contrast quantization

• **The visual perception of the luminance is non-linear, *but the visual perception of contrast is linear***
⇒uniform quantization of the contrast is better than uniform quantization of the brightness
⇒*contrast* = ratio between the lightest and the darkest brightness in the spatial region
⇒just noticeable changes in contrast: 2% => 50 quantization levels needed ⇔ 6 bits needed with a uniform quantizer (or 4-5 bits needed with an optimal quantizer)





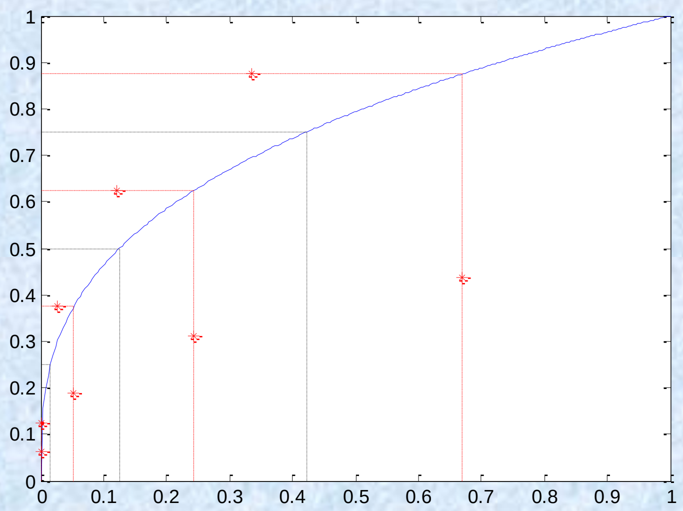
$$c = \alpha \ln(1+\beta u), 0 \le u \le 1; \text{ typ. } \alpha = 6 \ldots 18, \alpha = \beta / \ln(1+\beta)$$

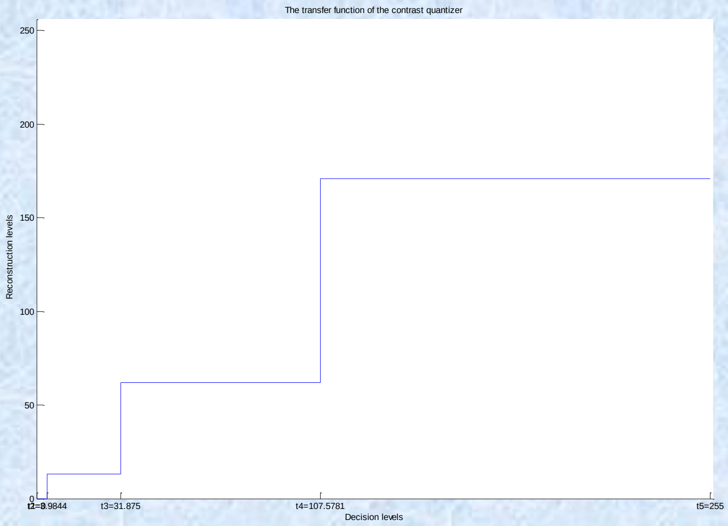
or

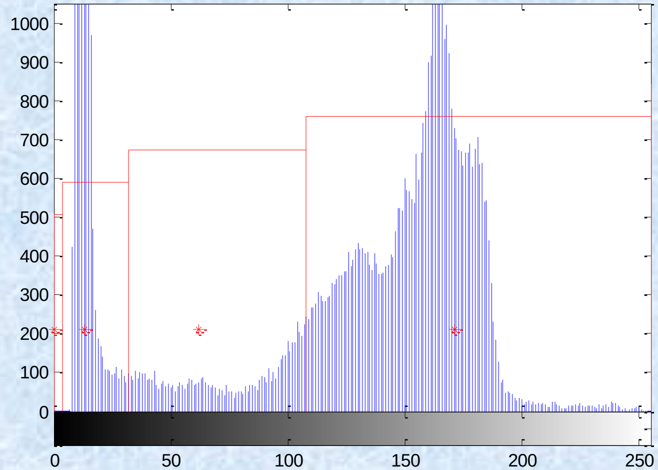
$$c = \alpha u^{\beta}; \text{ typ. } \alpha = 1; \beta = 1/3$$

Examples of contrast quantization:

- **For $c=u^{1/3}$:**

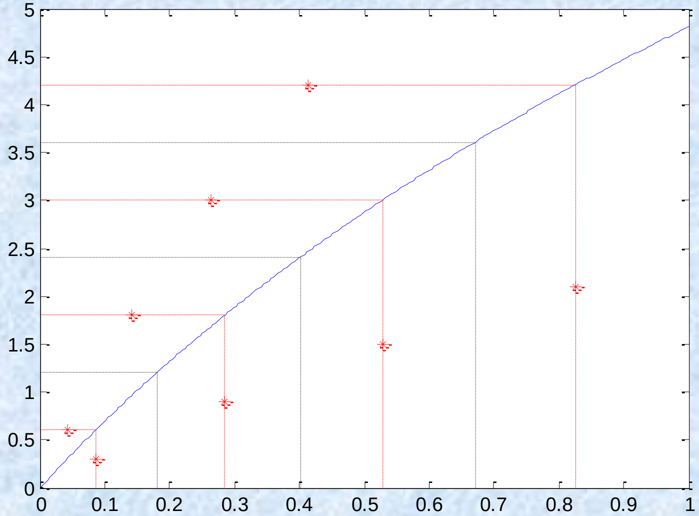


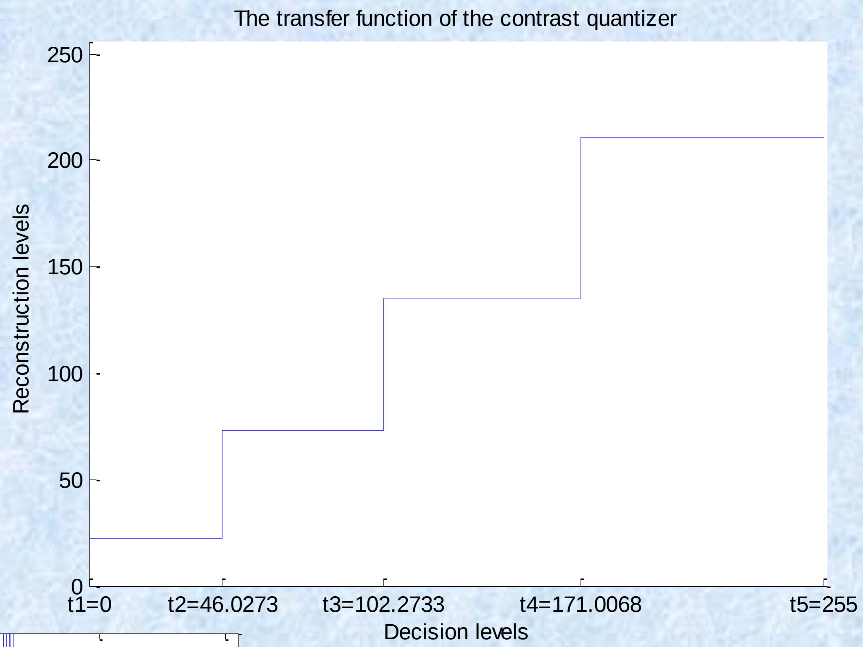


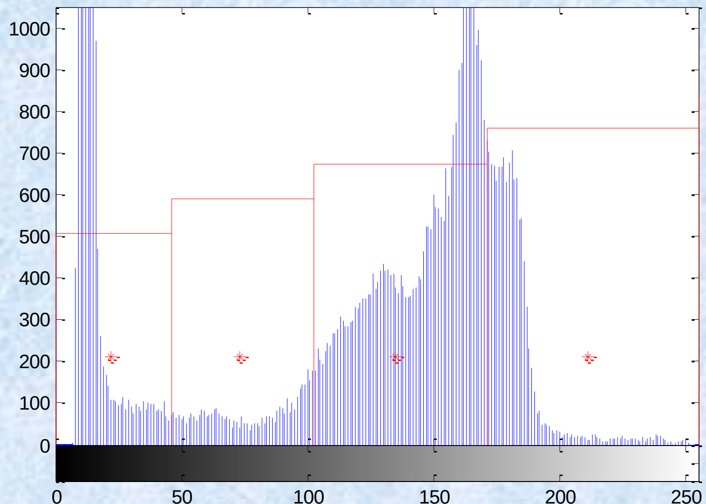


Examples of contrast quantization:

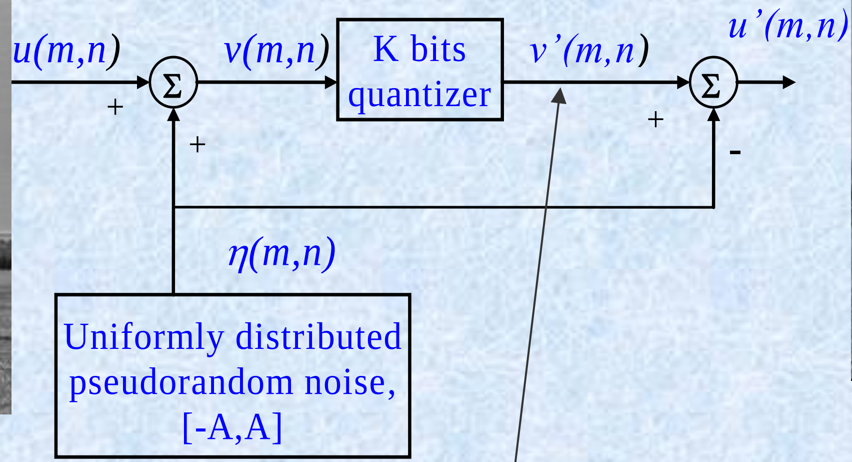
- **For the log transform**:





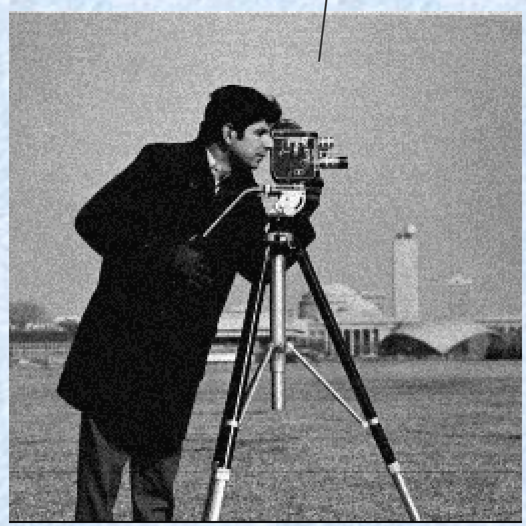


b. Pseudorandom noise quantization (“dither”)



Large dither amplitude

Uniform quantization, B=4



Prior to dither subtraction

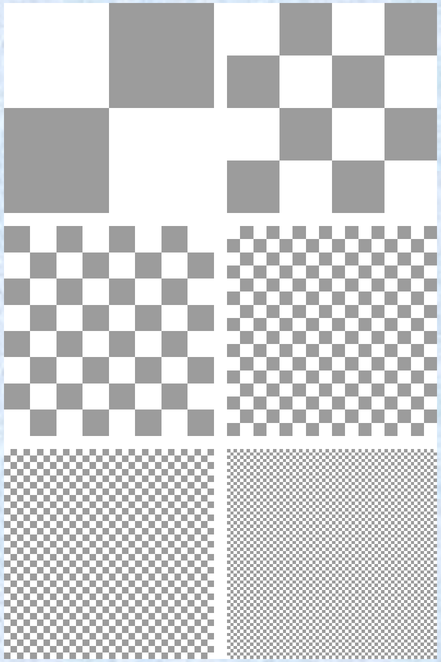
Small dither amplitude

Fig. 13

a b
c d

a. 3 bits quantizer =>visible false contours;
b. 8 bits image, with pseudo-random noise added in the range [-16,16];
c. the image from Figure b) quantized with a 3 bits quantizer
d. the result of subtracting the pseudo-random noise from the image in Figure c)

Halftone images generation



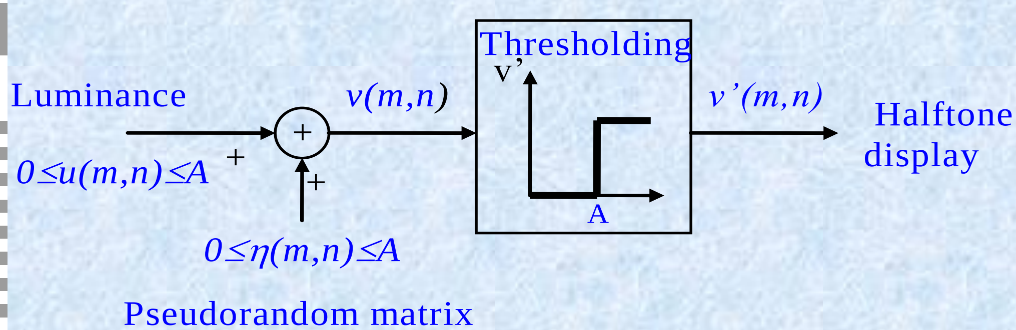


Fig.14 Digital generation of halftone images

$$H_1 = \begin{bmatrix} 40 & 60 & 150 & 90 & 10 \\ 80 & 170 & 240 & 200 & 110 \\ 140 & 210 & 250 & 220 & 130 \\ 120 & 190 & 230 & 180 & 70 \\ 20 & 100 & 160 & 50 & 30 \end{bmatrix}$$

$$H_2 = \begin{bmatrix} 52 & 44 & 36 & 124 & 132 & 140 & 148 & 156 \\ 60 & 4 & 28 & 116 & 200 & 228 & 236 & 164 \\ 68 & 12 & 20 & 108 & 212 & 252 & 244 & 172 \\ 76 & 84 & 92 & 100 & 204 & 196 & 188 & 180 \\ 132 & 140 & 148 & 156 & 52 & 44 & 36 & 124 \\ 200 & 228 & 236 & 164 & 60 & 4 & 28 & 116 \\ 212 & 252 & 244 & 172 & 68 & 12 & 20 & 108 \\ 204 & 196 & 188 & 180 & 76 & 84 & 92 & 100 \end{bmatrix}$$

Demo: http://markschulze.net/halftone/index.html

Fig.15 Halftone matrices

Fig.3.16

Color images quantization

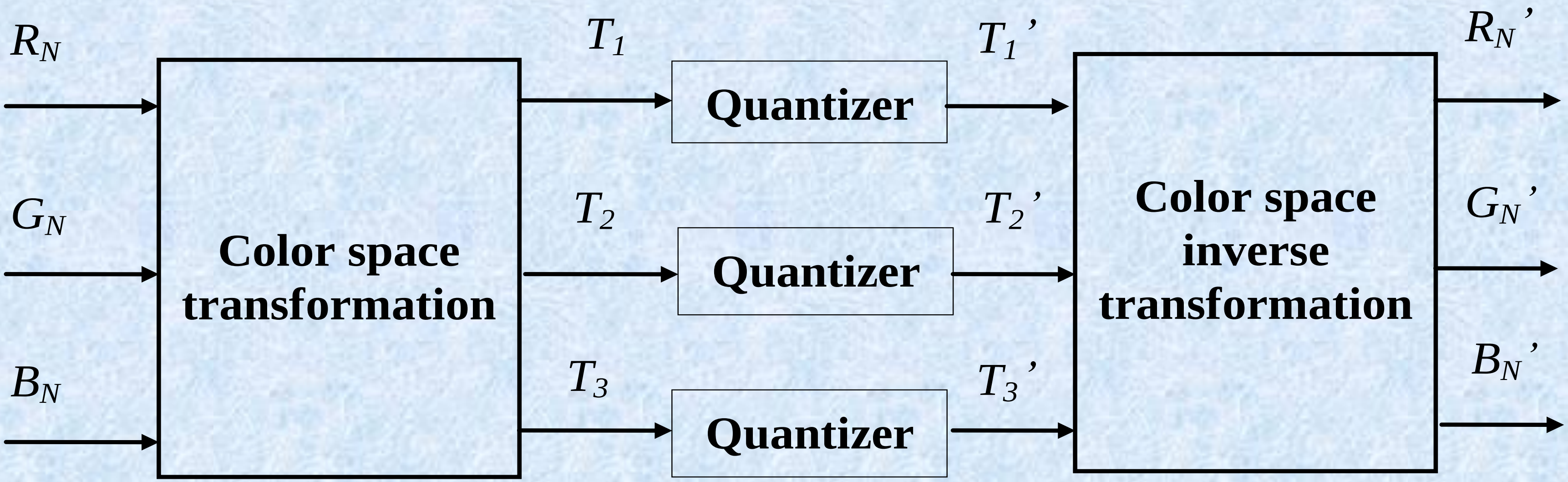


Fig.17 Color images quantization