

## 1. Přednáška: Signály a jejich reprezentace, kvantizace

### 1 Signály a některé jejich vlastnosti

#### 1.1 Klasifikace signálů (související pojmy)

*Pojem signál:*

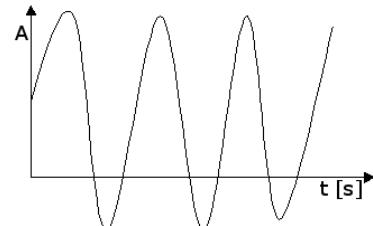
- fyzikální vyjádření informace, která je funkcí jedné nebo více nezávislých proměnných
- je funkcí času (prostoru)

*Nositelem inf.*

- mohou být pouze ty signály, které nemohou být na straně příjemce predikovány. Mají náhodný charakter.

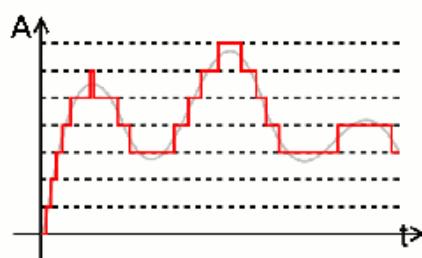
*Analogový signál:*

- signál spojitý v amplitudě i v čase. Může nabývat libovolné hodnoty z určitého spojitého intervalu možných hodnot. Jeho typickou vlastností je *reprezentace E => reálně neexistují nespojitosti!*

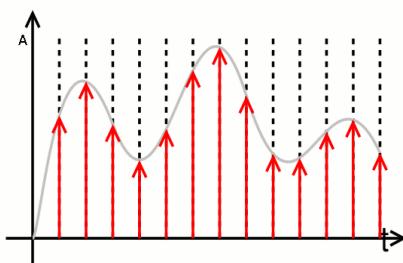


*Diskrétní signál (pouze vyjádření skutečného signálu!):*

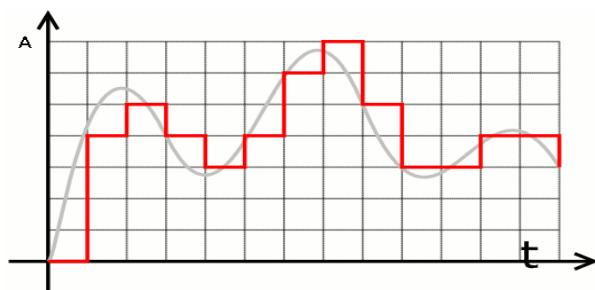
- na rozdíl od spojitého signálu se vyznačuje určitým druhem nespojitosti
- *kvantovaný signál* - signál diskrétní v amplitudě, amplituda se mění po skocích, nabývá omezeného počtu stavů



- *vzorkovaný signál* - signál diskrétní v čase, tvořen posloupností vzorků (nabývajících libovolných hodnot)
  - počet vzorků za 1s udává vzorkovací kmitočet
  - nejčastěji konstantní krok – ekvidistantní vzorkování



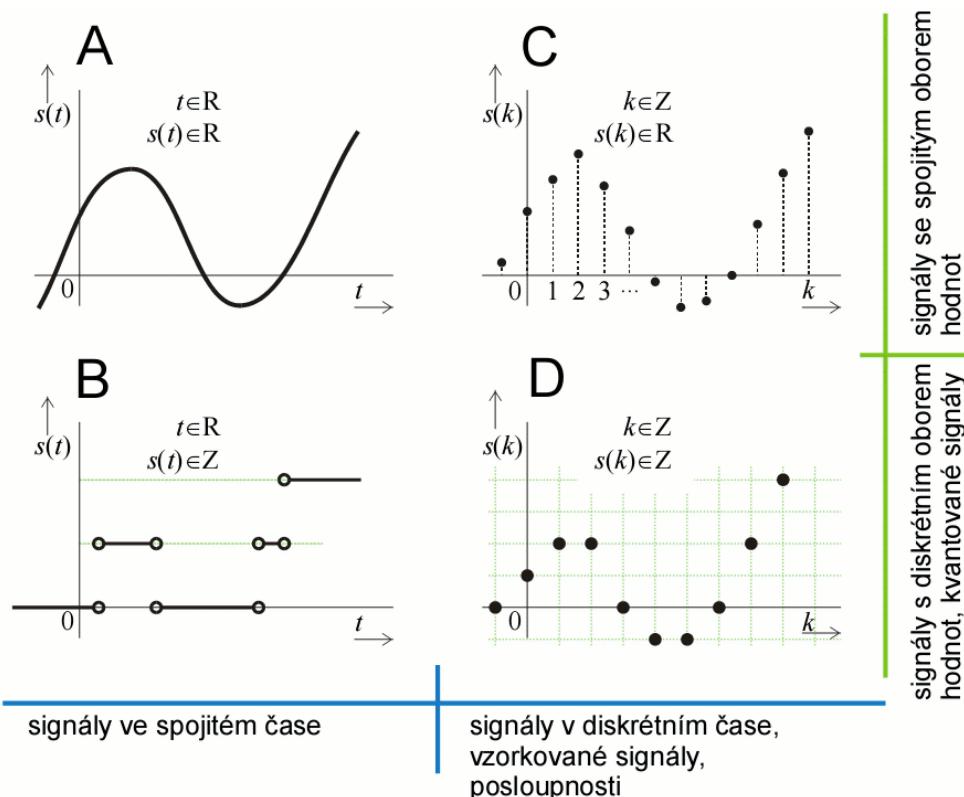
- **digitální signál** – vzorkovaný + kvantovaný signál
  - tvořen posloupností vzorků nabývajících omezeného počtu stavů



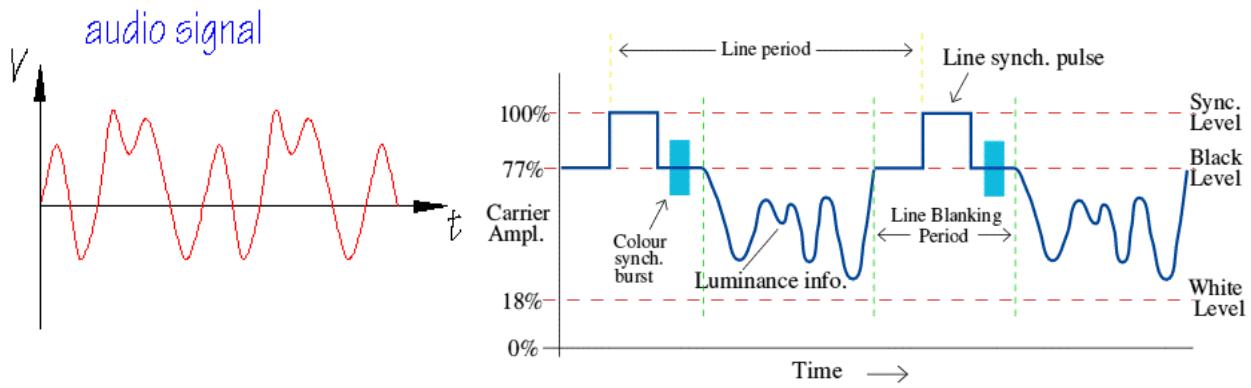
Pro signály s diskrétní časovou osou se vžil název vzorkované signály. Při diskrétní svislé ose hovoříme o signálech s kvantovanými hodnotami.

Má li signál obě osy spojité => [analogový signál](#).

Má li signál obě osy diskrétní => [digitální \(číslicový\) signál](#)

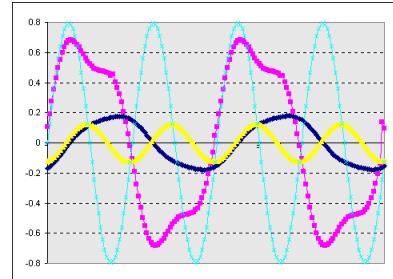


### Jednorozměrný X vícerozměrný signál



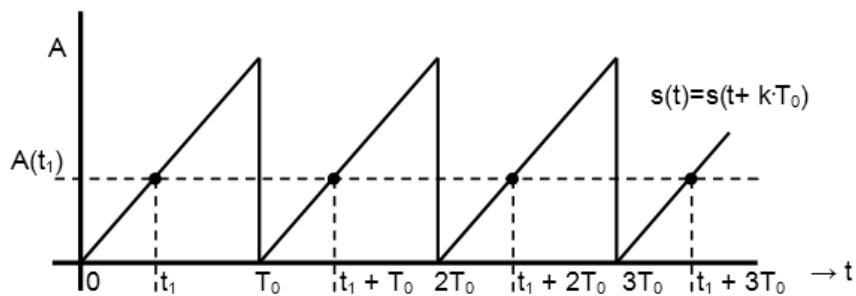
### Model signálu:

- popisuje reálný signál zjednodušenými parametry (konečný časový, frekvenční rozvoj...)
  - používán především při analýze soustav
- $$u(t) = 10 \sin(t + 0.9)$$
- Ne všechny signály je možné dobře modelovat!
    - typicky šum



### Periodický signál

$$s(t) = s(t + k \cdot T_0)$$



Matematická definice na intervalu  $(-\infty, \infty)$  - uvažuje nekonečný signál => **periodický signál je prakticky nevyrobiteLNÝ**.

- okrajová omezení (časová oblast i spektrum)
- pojmy stř./ef. hodnota se vztahují k části kterou považujeme za periodickou

Termín	Označení a jednotka
Opakovací perioda	$T_0$ [s]
Opakovací kmitočet	$F_0 = 1/T_0$ [Hz = s <sup>-1</sup> ]
Kruhový opak. kmitočet	$\omega_0 = \frac{(2\pi)}{T_0} = 2\pi F_0$ [s <sup>-1</sup> ]
Pulz	
Střední hodnota	$S_{stř}$
Efektivní hodnota	$S_{ef}$ nebo jen S

### Aperiodický signál:

- nesplňuje rov.  $s(t) = s(t+k \cdot T_0)$
- některé druhy sig. *pulzy*



- pro tyto signály mnoho jiných charakteristik, E, spektrální hustota...
  - ve skutečnosti všechny reálné signály aperiodické

### Signály deterministické a stochasticke

pojem souvisí s poznatelností (určeností) v libovolném čase

### Deterministický signál

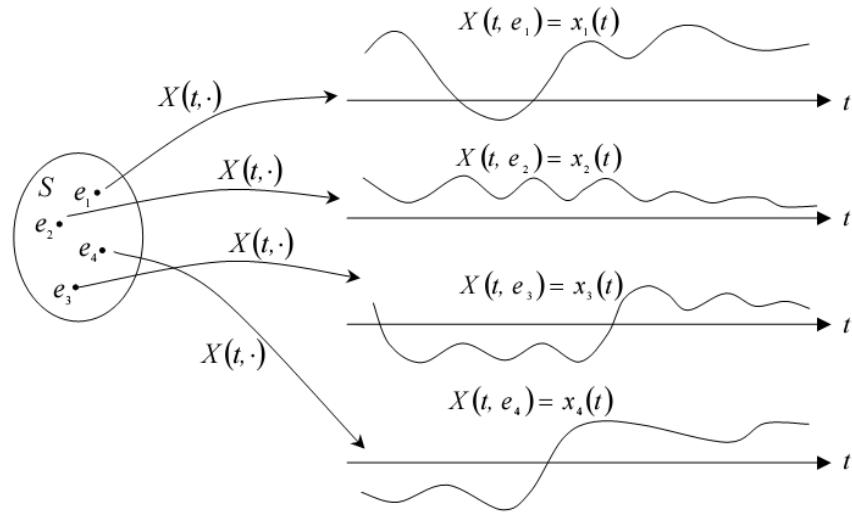
- signál přesně určený
- je možné vytvořit přesný časový (prostorový) model
- nese nulové množství *Inf.*
  - většina signálů nejsou deterministické – vytváříme zjednodušené modely
  - snaha o co největší přiblížení (nejmenší chybu)

### Stochastický signál

je generován stochastickým procesem,

nelze vytvořit věrný model

- je možné provést více měření, pokaždé s trochu jiným výsledkem => **množina realizací**

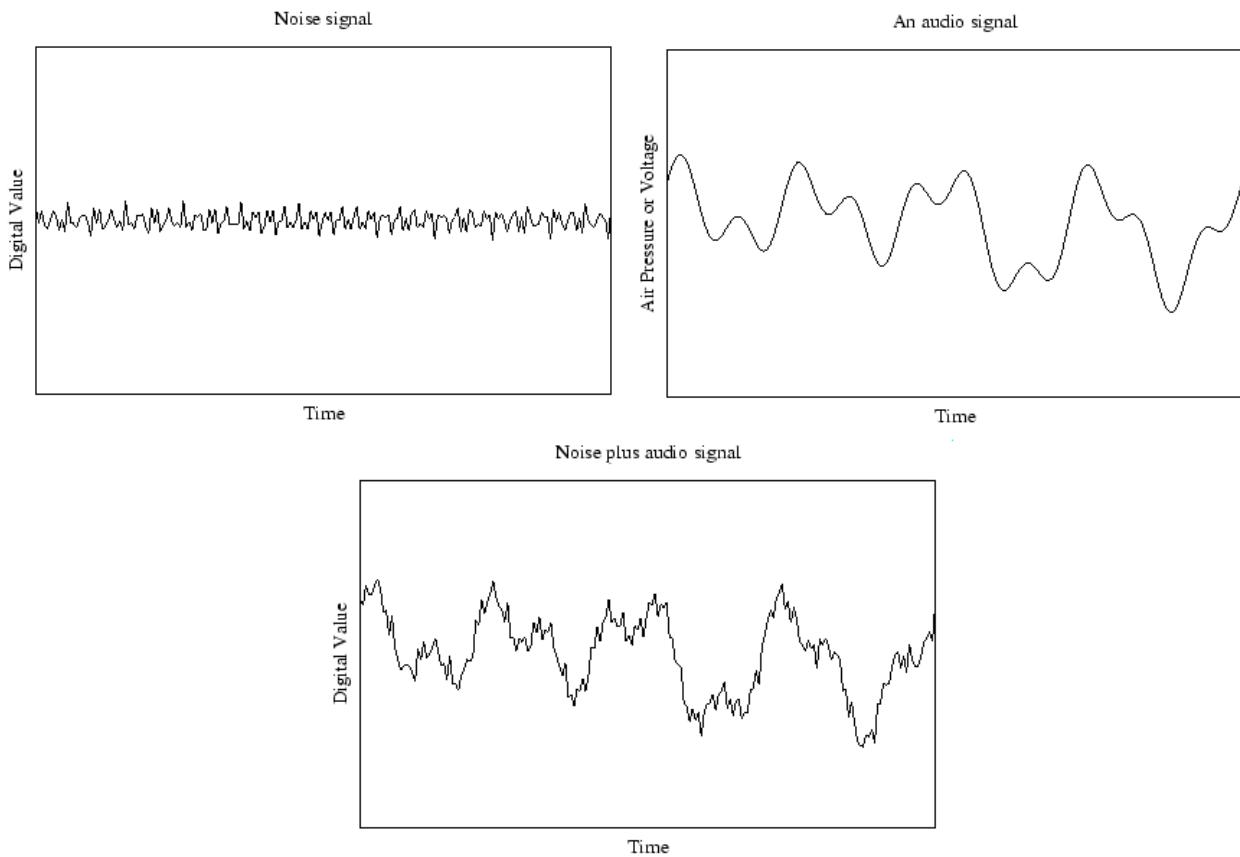


- má smysl pouze popis pomocí statistických charakteristik
  - (stř. hod., směrodat. odch, rozptyl, spektrální výkon....)

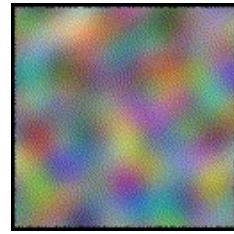
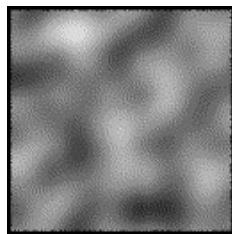
Typickým příkladem stochastického signálu je **šum**

V reálu je téměř každý zpracovávaný signál zatížen šumem

### **Šum v jednorozměrném signálu (jednorozměrný šum)**

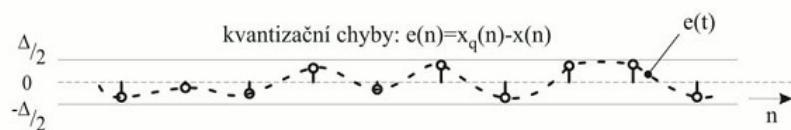
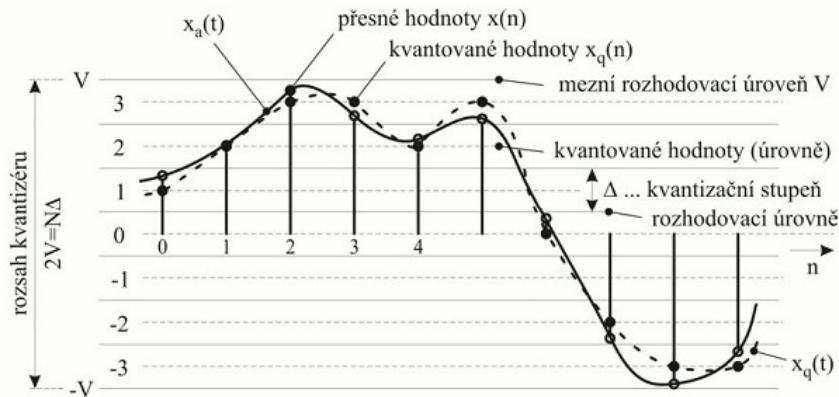


Šumem mohou být zatíženy i vícerozměrné signály (vícerozměrný / vícedimenzionální) šum



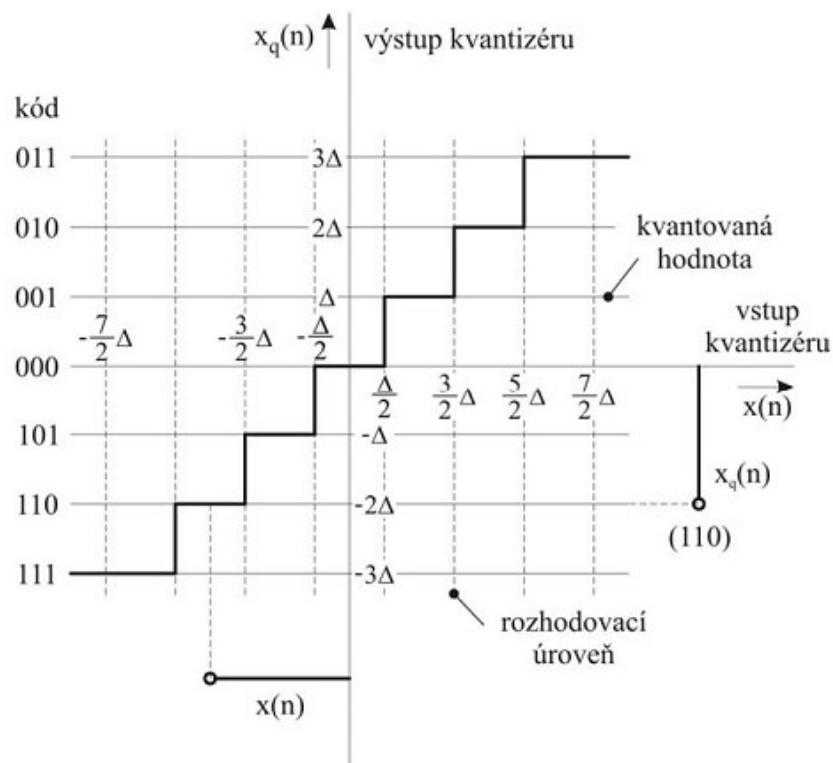
## 1 Kvantifikace amplitud

- kvantování  $x(n)$ <sup>{rozsahy D}</sup> →  $x_q(n)$ <sup>{konečný počet diskrétních hodnot}</sup>
    - lineární – kvantizační stupně mají jednotnou šířku  $D$
    - nelineární – typicky malé úrovně jemněji (audiosignál), snaha o zachování kvality při zkrácení kódového slova,
      - POZOR! - při zpracování např. dig. filtrem vždy převod na lineární kvantování a zpět → degradace signálu
    - vzniká kvantizační chyba:
- $$e(n) = x_q(n) - x(n)$$
- *náhodný charakter* v rozsahu  $-D/2 \leq e(n) \leq D/2 \rightarrow$  **kvantizační šum** (kvantizační zkreslení, zrnitý šum)
  - kvantizace – vždy degradace signálu (nelze zpětně přesně rekonstruovat původní signál)
- rozsah kvantizéru ±  $V_{max} = 2V_{max}$ 
    - při překročení – saturace (náhodný proces) – generuje **overload noise**, lze potlačit vhodnou volnou  $V_{max}$
    - rozsah typicky  $N = 2^b \rightarrow \Delta = \frac{2V}{N} = \frac{2V}{2^b}$  ( $b$  – počet bitů v kódových slovech),

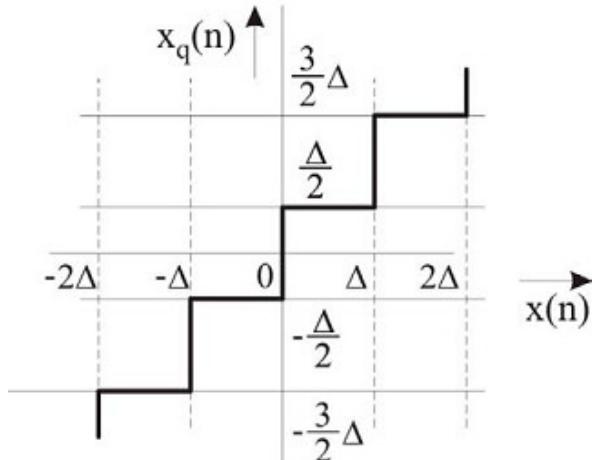


Princip kvantování a základní pojmy (Kvantování zaokrouhlováním)

- převodní charakteristika – vyjadřuje funkci kvantizéru
  - vyjadřuje rozhodovací úroveň → výstupní kód kodéru
  - *lineární kvantizér v okolí 0* (resp. každé rozhodovací úrovňě!) dvě možnosti
    - kvantizér s nulovou **kvantovací úrovní** – potlačuje vstupní rušivé amplitudy menší než  $\pm D/2$  – jsou kvantovány na 0, počet kv. stupňů  $N = 2^b - 1$ 
      - *to ale platí i pro užitečný signál*

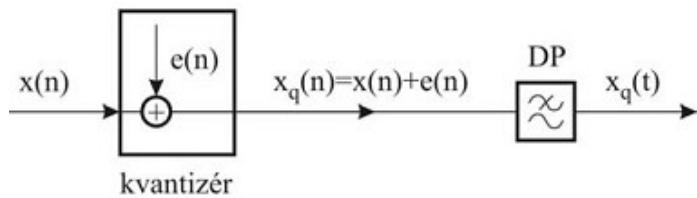


- kvantizér s nulovou **rozhodovací úrovni** – *autogenerující šum o  $\pm D/2$*  (i bez  $Sig_{IN}$  generuje  $Sig_{OUT}$  náhodně přeskakující mezi  $\pm D/2$  (*klidový šum kvantizéru*), počet kvantizačních stupňů  $N = 2^b$ )

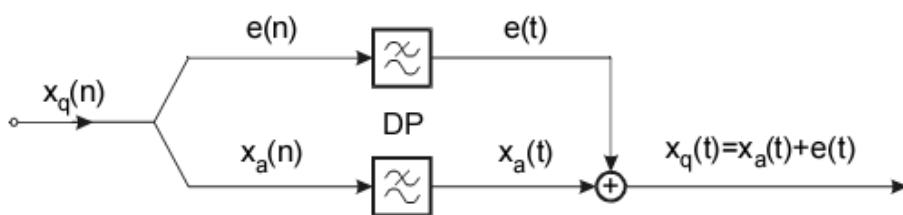


*Jak ovlivňuje kvantizační šum užitečný signál?*

- model rekonstrukce signálu podle  $e(n) = x_q(n) - x(n)$



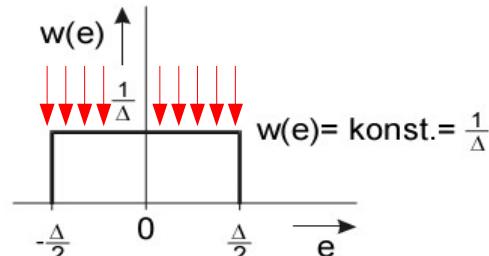
- $x_q(t)$  – obnoveny analogový signál zatížený *kvantizační chybou*  $e(n)$  (kvantizačním šumem)
  - *kvantizační chyba*:  $e(n) = x_q(n) - x(n) \rightarrow$  lineární systém (superpozice) → lze oddělit (zpracovávat nezávisle)
- při předpokladu: chyby  $e(n)$  nastávají náhodně - nekorelovaný  $e(n)$  s  $x(n)$ 
  - provedeme-li rekonstrukci:



- potom je možné posuzovat nezávisle – resp. jako poměr
- obvyklé měření např. výkonový poměr signál / šum  $D = 10 \log \frac{\sigma_x^2}{\sigma_e^2} [dB]$
- (nebo odstup signál – šum)  $D_s = 20 \log \frac{\sigma_x}{\sigma_e} [dB]$ 
  - $s^2_x$  – disperze signálu (při zátěži 1W = výkon)

Jak určit výkon šumu? -  $s^2_e$

- uvažujme:
  - kvantizační chyby – projevují se náhodně, jako náhodný šum
    - → lze kvantifikovat pouze přes statistické parametry
  - předpoklad – chyby  $e(n)$  jsou nekorelované se signálem (nastávají v rámci  $\pm D/2$  náhodně)
    - pak rovnoramenné rozdělení hustoty pravděpodobnosti

Rozdělení hustoty pravděpodobnosti chyb  $e_a(n)$ 

Potom  $s^2_e$ :

$$\sigma_e^2 = E[e^2(n)] = \int_{-\Delta/2}^{\Delta/2} e^2 w(e) de = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2 de = \frac{1}{\Delta} \left[ \frac{e^3}{3} \right]_{-\Delta/2}^{\Delta/2} = \frac{\Delta^2}{12}$$

pro rozsah  $2V$  a počet kvantizačních stupňů  $N=2^b$ , tedy  $\Delta=\frac{2V}{2^b}$   $\rightarrow \sigma_e^2 = \frac{\Delta^2}{12} = \frac{V^2}{3 \cdot 2^{2b}}$   $\rightarrow \sigma_e = \frac{V}{\sqrt{3} \cdot 2^b}$

- máme výkon šumu – je závislý na počtu kvantizačních stupňů (hloubce A/D převodníku) a rozsahu převodníku

....je možné vyjádřit *odstup signál / šum*:

$$D_s = 20 \log \frac{\sigma_x}{\sigma_e} [dB] = 20 \left\{ \log \sigma_x - \log \frac{V}{\sqrt{3} \cdot 2^b} \right\} = 20 \left[ \log \sigma_x - (\log V - \log \sqrt{3} \cdot 2^b) \right] = \dots$$

$$\dots = -20 \log \frac{V}{\sigma_x} + 4,77 + 6,02 b [dB]$$

Odstup signál / šum závisí na použitém A/D převodníku

- obvykle se měří např. harmonickým signálem  $\rightarrow x(t) = A \cos(\omega t)$

- potom ef. hodnota  $\sigma_x = x_{ef} = \frac{A}{\sqrt{2}}$

a pro měřící harmonický signál  $x(t) = A \cos \omega t$  s rozkmitem  $A = V \rightarrow \sigma_x = e_{ef} = A/\sqrt{2}$

$$D_s = 20 \log \frac{\sigma_x}{\sigma_e} = 4,77 - \log \sqrt{2} + 6,02 b - 20 \log \frac{V}{A} = 1,76 + 6,02 b - 20 \log \frac{V}{A} [dB]$$

$$D_s = 20 \log \frac{\sigma_x}{\sigma_e} = 1,76 + 6,02 b [dB]$$

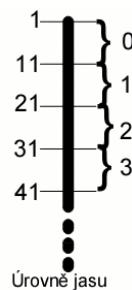
- je vidět, že kvalita závisí na velikosti kódového slova  $b$  resp. počtu kv. stupňů  $N = 2^b$

- množství šumu můžeme ovlivnit volbou  $D$ , resp  $b \rightarrow$  narůstá délka kódového slova
- je možné i opačně  $\rightarrow$  [pro požadovanou kvalitu zvolit vhodný A/D převodník](#)
- každé rozšíření převodníku o  $1b \rightarrow$  zvětšení  $D_s$  o  $6 \text{ dB}$

Příklady:

- pro 4 bitový audiozáznam  $D_s = 1,76 + 6,02 \cdot 8 = 26 \text{ dB}$
- pro 8 bitový audiozáznam  $D_s = 1,76 + 6,02 \cdot 8 = 50 \text{ dB}$
- pro 16 bitový audiozáznam  $D_s = 1,76 + 6,02 \cdot 16 = 98 \text{ dB} \rightarrow$  např. kvalita CD

počet  $k$  možných hodnot omezen bitovou hloubkou  $b$  použitých prostředků  $k = 2^b$



**Příklad 1:** kvantifikace intenzity audiosignálu

[mp3 příklad](#)

**Příklad 2:** kvantifikace intenzity světla

lidské oko  $\rightarrow$  cca 100 úrovní jasu

$2^5 = 32$  jasových úrovní

$2^8 = 256$  jasových úrovní





Originál 256 jasových úrovní



4 jasové úrovně

## Digitalizace signálu

- Digitalizace – převod  $AS \rightarrow DS$  (analogový → diskrétní  $\xrightarrow{\text{v čase}} \xrightarrow{\text{v amplitudě}}$  signál )
- Obvykle převod spojité předlohy (reality)  $\rightarrow f_1(t/x, \dots), f_2(\dots)$

připomenutí Digitalizace:

1. **vzorkování** (časové / prostorové rozlišení)
2. kvantizace (amplitudy, hodnoty barev...)

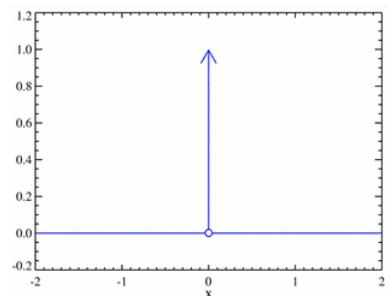
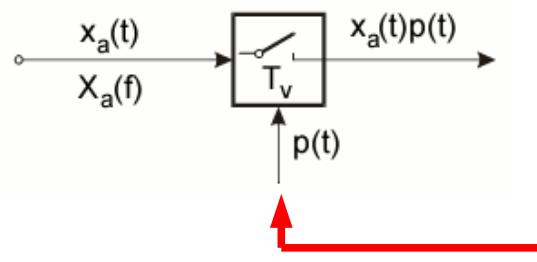
### Vzorkování

- odebírání vzorků v časových intervalech  $T_{vz}$  [s] se vzorkovací rychlosí  $F_V = 1 / T_{vz}$  [vzorků / s].
  - obvykle  $T_{vz} = \text{konst.} \rightarrow \text{ekvidistantní vzorkování}$

Ideální vzorkování – **neomezeně úzké** vzorkovací impulzy – proč?

Lze modelovat:

- násobením  $x_a(t)$  vhodnou vzorkovací funkcí  $p(t)$
- jak vybrat vhodnou vzorkovací funkci?



- neomezeně úzký impulz =  $\delta(x) = \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$

Diracův pulz má některé důležité časové / frekvenční vlastnosti:

Důležité vlastnosti  $d(t)$ :

$$\int_{-\infty}^{\infty} x_a(t) \delta(t - t_0) = x_a(t_0) \quad (1)$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

$$\int_{-\infty}^{\infty} f(x) \delta(x) dx = f(0)$$

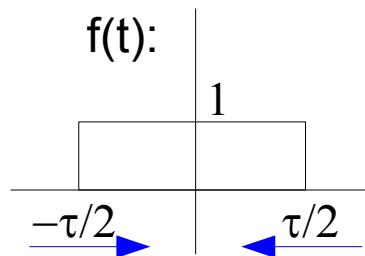
$$x_a(t)|_{t=t_0} = x_a(t) \delta(t - t_0) = x(t_0) \delta(t - t_0) \quad (2)$$

Jak spočítat spektrum Diracova pulzu?:

Diracův pulz je možné modelovat jako

„zužující“ se jednotkový pulz (limitně k šířce = 0)

$d(t)$  je možné „vyrobít“  $d(t) = A \cdot 1(t)$ , kde  $A \rightarrow \infty$

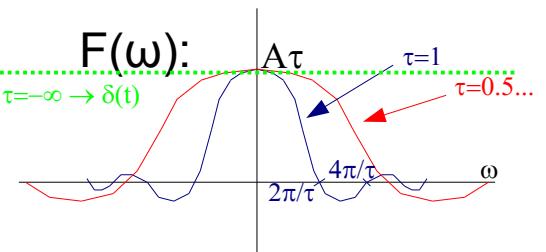


potom:

$$F[\delta(t)] = \int_{-\tau/2}^{\tau/2} A \cdot 1 \cdot e^{-j\omega t} dt = A \cdot \left[ \frac{e^{-j\omega t}}{-j\omega} \right]_{-\tau/2}^{\tau/2} = A \cdot \frac{e^{-j\omega\tau/2} - e^{j\omega\tau/2}}{-j\omega} = A \cdot \frac{+e^{j\omega\tau/2} - e^{-j\omega\tau/2}}{j\omega} \cdot \frac{2}{\omega} = \dots$$

$$\dots = A \cdot \frac{\sin\left(\omega \frac{\tau}{2}\right)}{\omega \cdot \frac{1}{2}} \cdot \frac{\tau}{\tau} = \dots = A \cdot \tau \cdot \frac{\sin(x)}{x} \quad \lim \tau \rightarrow 0 = 1$$

Red arrow pointing right.



Nebo pomocí představy: „vyseknutí“ signálu v místě  $t_0$  – rov. posunu  $d(t)$  (1):

$$F[\delta(t)] = \int_{-\infty}^{\infty} \delta(t) e^{(-j\omega t)} dt = \dots \quad |rov. (2): \int_{-\infty}^{\infty} x_a(t) \delta(t - t_0) dt = x(t_0)|_{t_0=0} \dots = \frac{e^{(-j\omega 0)}}{(-j\omega)} = 1$$

Velmi důležitý poznatek

spektrum  $d(t)$  obsahuje rovnoměrně zastoupeny všechny kmitočty (a to i záporné!)

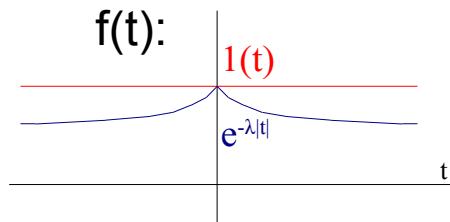
- takový signál nelze vyrobít!
- paradoxně nejkratší pulz → energeticky nejnáročnější
- nikdy nelze vyrobit např. pulzy s nekonečně krátkou vzestupnou / sestupnou hranou!
- reálně vždy vzorkujeme nedokonalou  $p(t)$

A obráceně – spektrum konstantního signálu?:

$$F\{1(t)\} = \int_{-\infty}^{\infty} 1 \cdot e^{-j\omega|t|} dt = \left[ \frac{e^{-j\omega|t|}}{-j\omega} \right]_{-\infty}^{\infty}$$

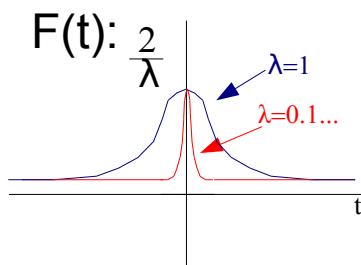
problém:  $e^{j\omega|t|}$  nemá pro  $t \rightarrow \pm\infty$  limitu

proto náhrada  $1 \rightarrow e^{-\lambda|t|}$



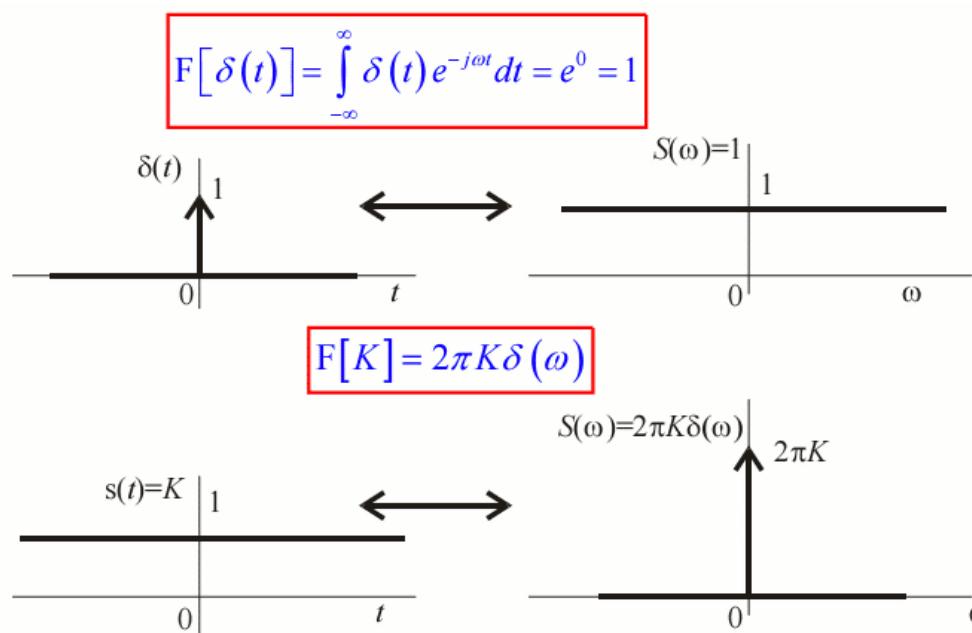
$$F(\omega) = \int_{-\infty}^{\infty} e^{-\lambda|t|} \cdot e^{-j\omega t} dt = \int_{-\infty}^0 e^{\lambda t} \cdot e^{-j\omega t} dt + \int_0^{\infty} e^{-\lambda t} \cdot e^{-j\omega t} dt = \left[ \frac{e^{(\lambda-j\omega)t}}{\lambda-j\omega} \right]_0^0 + \left[ \frac{e^{-(\lambda+j\omega)t}}{-(\lambda+j\omega)} \right]_0^{\infty} = \dots$$

$$\dots = \frac{1}{\lambda - j\omega} - 0 + 0 + \frac{1}{\lambda + j\omega} = \frac{2\lambda}{\lambda^2 + \omega^2} \quad \lim \lambda \rightarrow 0 = K \cdot \delta(\omega)$$



Důležitý poznatek:

- jediná spektrální čára by vyžadovala nekonečný signál (v čase)!
- širokopásmové rušení je možné potlačit pulzními zásahy do signálu
- má-li signál nenulovou ss složku  $\rightarrow$  spektrum obsahuje  $d(t)$  na nulovém kmitočtu



**Vzorkování = vynásobení signálu  $x(t)$  s nekonečnou řadou  $d(t)$ .**

Výsledek konvoluce spekter:

za použití (2):

$$x(t) = x_a(t) p(t) = x_a(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_v) = \sum_{n=-\infty}^{\infty} x(nT_v) \delta(t - nT_v) \quad (3)$$

a z toho spektrum:

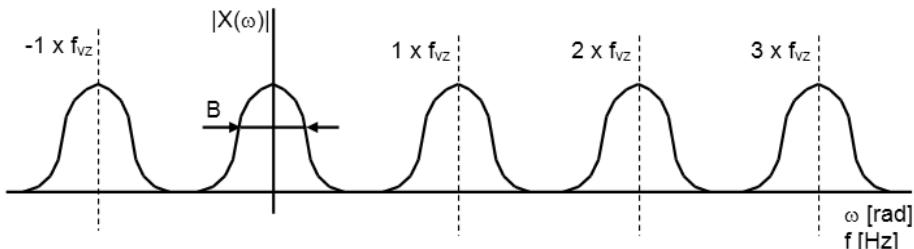
$$X(\omega) = F[x_a(t)p(t)] = F\left[x_a(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_v)\right] = \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x_a(t) \delta(t - nT_v) e^{-j\omega t} dt \quad (4)$$

s uvážením rov. (1) má pak integrál řešení:  $x(nT_v) e^{-j\omega nT_v} \rightarrow$  spektrum **JEDNOHO VZORKU**

Pro diskrétní posloupnosti pak podle (4):

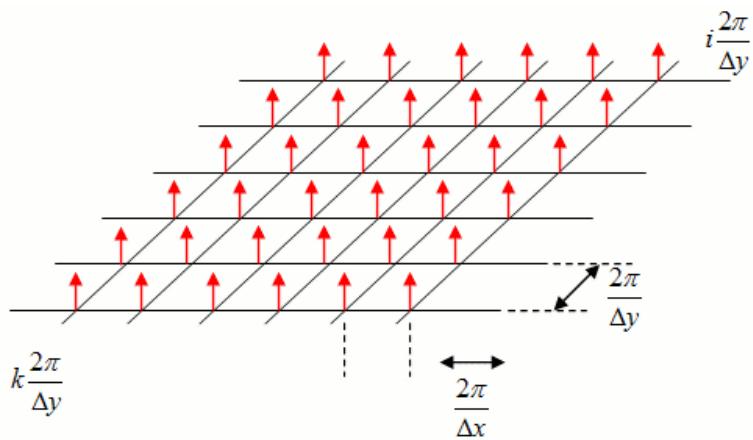
$$X(\omega) = \sum_{n=-\infty}^{\infty} x(nT_v) e^{(-j\omega nT_v)} \quad (5)$$

*Spektrum diskrétní posloupnosti (= vzorkovaného signálu) je spojité a periodické s periodou  $f_{vz} = 1/T_v$  = obsahuje spektrum původního analogového signálu posazené (zkopírováné) na  $f_{vz}$ .*



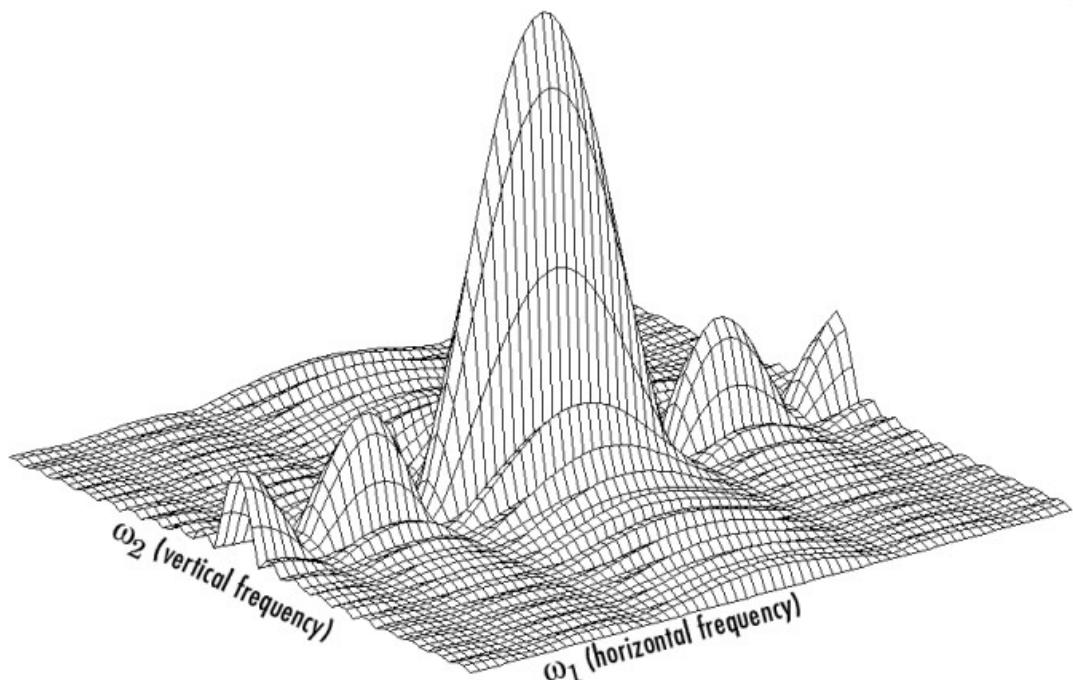
**Spektrum vícerozměrných signálů**

Spektrum 2D signálu se získá pomocí 2D vzorkovací funkce:



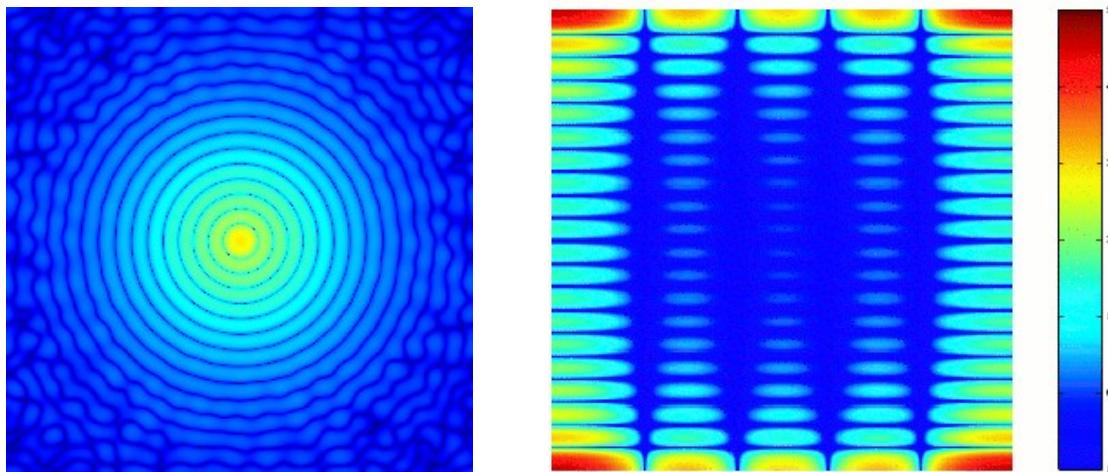
$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x - x_0, y - y_0) dx dy = f(x_0, y_0)$$

Výsledkem vzorkování je 2D spektrum signálu



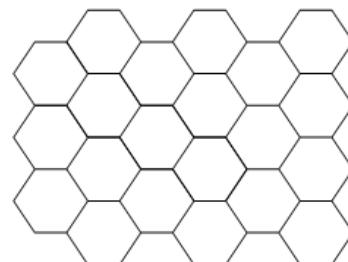
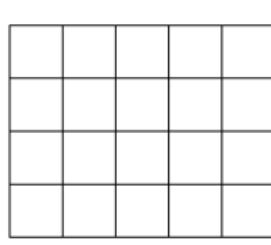
- podobně jako v případě 1D vzorkování se signál bude ve frekvenci opakovat na „nosných“

Pro zobrazení 2D FT se velmi často používá zobrazení v ploše (více v dalších přednáškách):

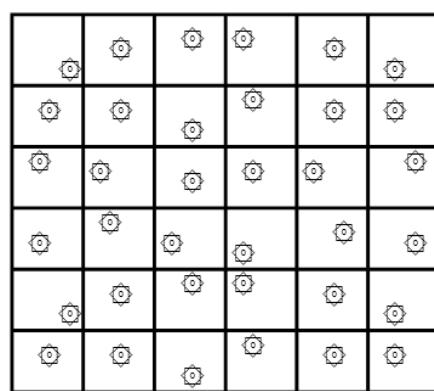


Pro vlastní vzorkování se nejčastěji používají systémy ekvidistantního vzorkování:

- nejčastěji se čtvercovou mřížkou (počítače, dig. tech.)
- nebo systémy s hexagonální mřížkou (analogová technika, snímací kamery)



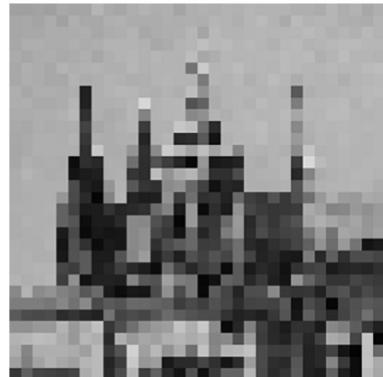
Ve výjimečných případech se používají např. systémy se stochasticky rozmístěnými D funkcemi.



Obraz vzorkovaný maticí  $256 \times 256$  a  $32 \times 32$  obrazových bodů:



Originál  $256 \times 256$



$32 \times 32$

## Aliasing

Při vzorkování platí [Nyquistovo kritérium](#) (Shannonův vzorkovací teorém):

$$f_v > 2 f_{sig} \quad (\dots v \text{ obrázku} = B)$$

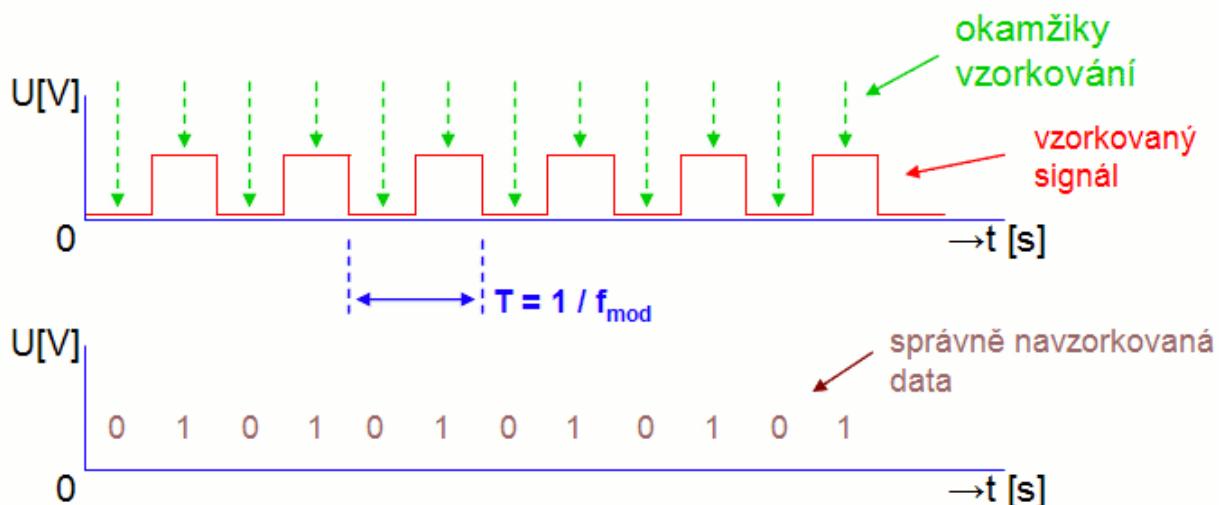
- Vzorkovací frekvence musí být min dvojnásobkem maximální fr. obsažené v datech
- Nemá cenu vzorkovat vyšší než dvojnásobnou fr. data s maxim. frekv.  $f_{sig}$ !

Co se stane pokud nedodržíme:

dva pohledy:

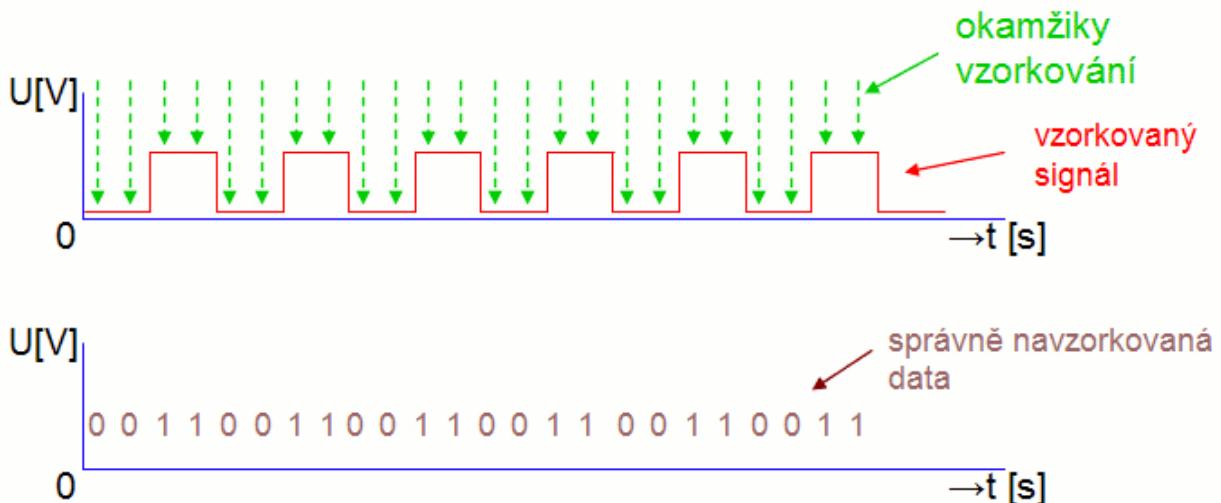
### 1. časová oblast

Mějme signál s modulační frekvencí  $f_{mod}$  který budeme vzorkovat podle Nyquistova kritéria  $f_v = 2 \times f_{sig}$ :



Co se signálem stane, pokud budeme vzorkovat rychleji?

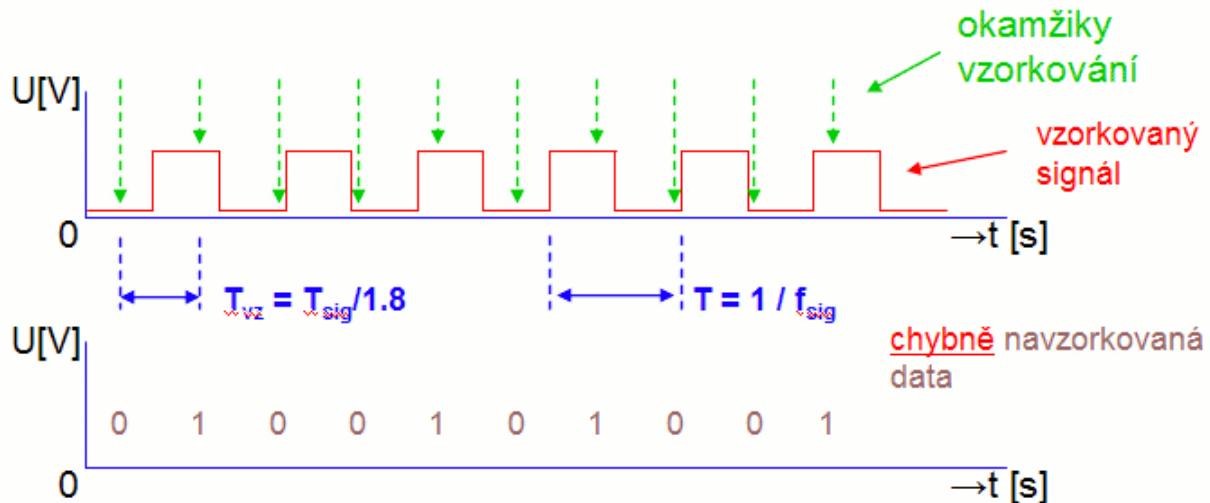
např. se 4 násobnou frekvencí ( $T_n = T/4$ )



- je vidět, že jsme získali znovu stejný průběh
  - pouze máme dvojnásobné množství hodnot, které ho popisují, to ale nevadí, spíše naopak
  - když budeme vzorkovat 10x rychleji, už budou 1ky a 0ky pěkně popisovat průběh signálu
  - žádnou novou inf. však nezískáme!

Co se ale stane, když tentýž signál budeme vzorkovat pomaleji?

například s rychlosí  $f_{vz} = 1.8 \times f_{sig}$

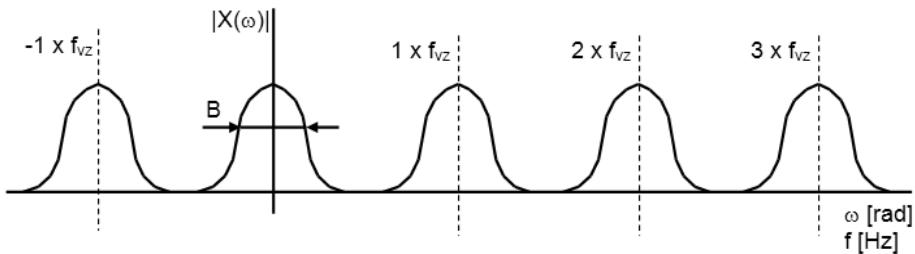


- je vidět, že při vzorkování menší než dvojnásobnou rychlosí došlo k podvzorkování
  - některá data se nám zcela ztratila
  - celkově došlo k poškození dat

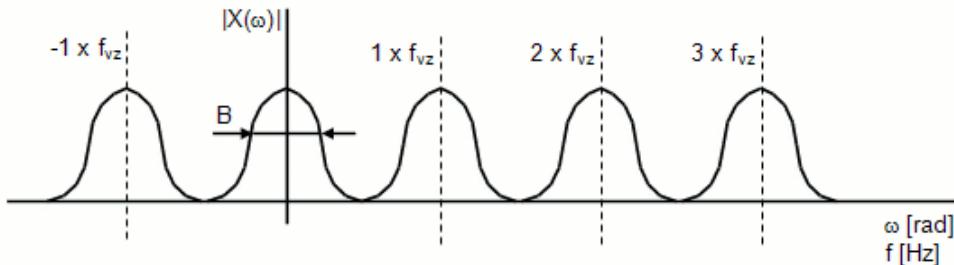
druhý pohled:

## 2. frekvenční oblast

Původní spektrum analogového signálu podle  $X(\omega) = \sum_{n=-\infty}^{\infty} x(nT_v) e^{(-j\omega T_v)}$  „sedí“ na násobcích vzorkovací frekvence. Pokud je  $f_{vz} \gg 2 \times f_{sig}$ :

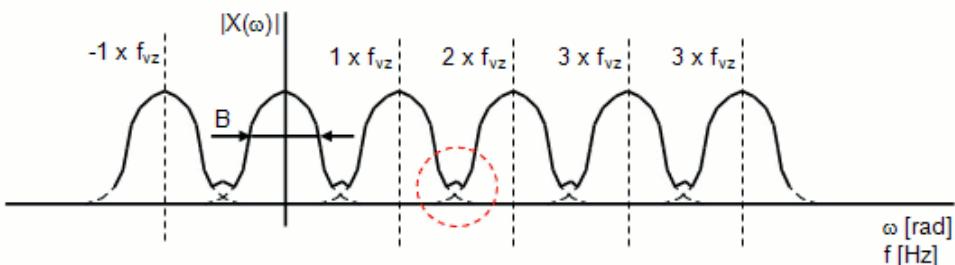


Co se se signálem stane, pokud zmenšíme  $f_{vz}$  na  $f_{vz} = 2 \times f_{sig}$ ?

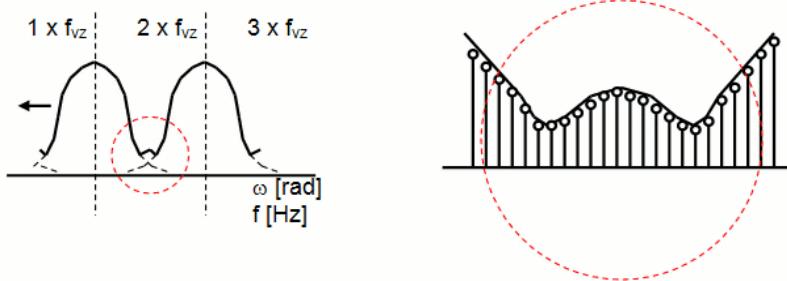


spektra se nepřekrývají ( $f_{vz} = 2 \times f_{sig}$ )

Co se se signálem stane, pokud zmenšíme  $f_{vz}$  na  $f_{vz} < 2 \times f_{sig}$ ?



Nejen, že došlo k ořezání spekter (např. 15% každého spektra se ztratilo), ale v této části došlo i k sečtení částí spekter → deformace (poškození) informace → signál je znehodnocen.



Jak se to projeví na reálném signálu?

původní inf. se poškodí – v signálu se objeví „alias“

- *alias* – vada v signálu vzniklá podvzorkováním
- *aliasing* – proces vzniku aliasů v důsledku nevhodného vzorkování
- *antialiasing* - je obecný pojem pro metody potlačování podvzorkování

příklad, obrazový signál:

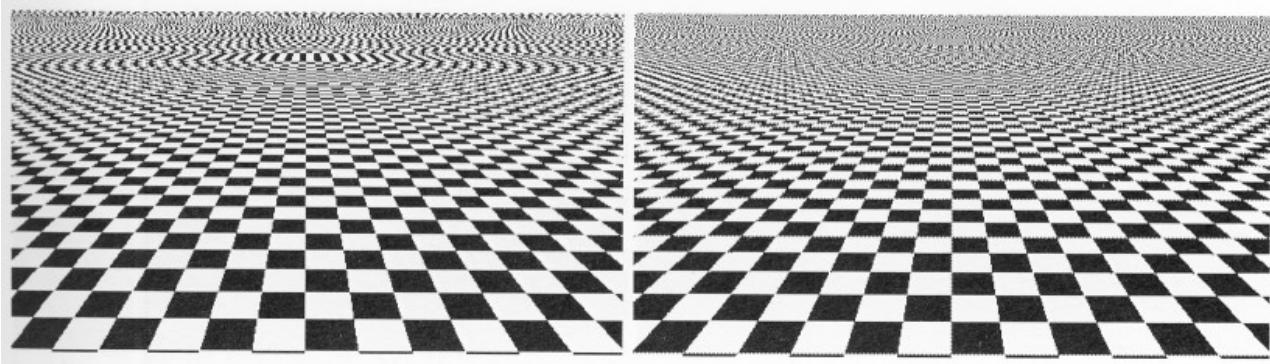


Některé signály mají teoreticky nekonečné spektrum, či se mu blíží.

Např. 2D obraz ideální fotografie.

→ proto ani nekonečným zvyšováním  $f_{vz}$  nelze aliasing zcela potlačit.

je pouze možné posunout oblast výskytu „aliasů“ do vyšších frekvencí:



### 3. Přednáška: Fourierova transformace, spektrum signálu

#### 1 Fourierova transformace

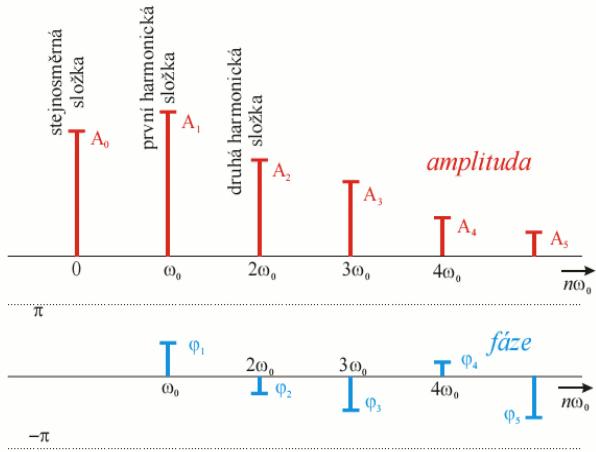
Proč FT? :

- základní matematický nástroj pro analýzu signálu
- převod mezi časovou a spektrální oblastí
- komprese inf. ...

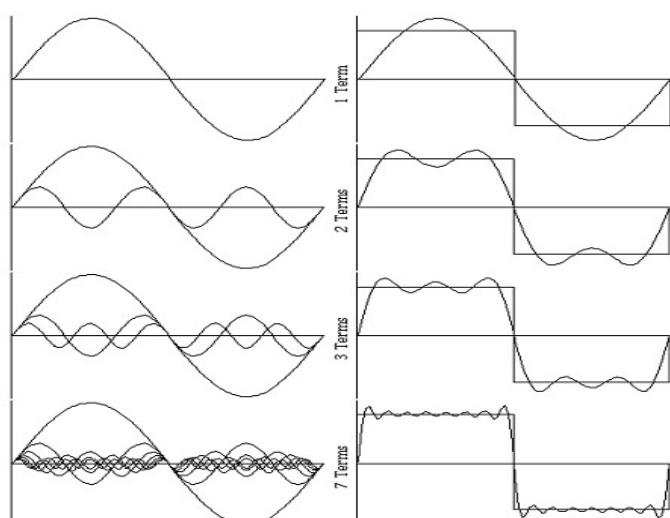
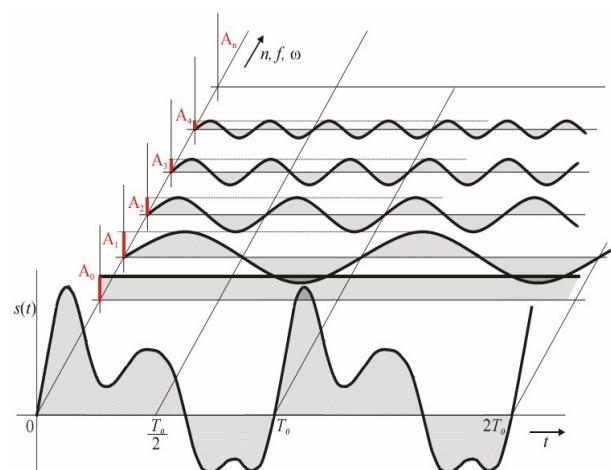
FT rozkládá analogový signál do **bázového prostoru**

**Bázový prostor** – tvořen kombinací sinusovek (kosinusovek – liší se pouze fázovým posunem → tedy posunem v časovém/předmětovém prostoru)

- **ortogonální systém** – každá sinusovka na násobku frekvence předchozí
  - představíte-li si je jako vektory → pak jejich vynásobením dostanete 0
- **úplný systém** – pokryty frekvence od  $-\infty$  do  $+\infty$



$$s(t) = A_0 + A_1 \cdot \cos(\omega_0 t + \varphi_1) + A_2 \cdot \cos(2\omega_0 t + \varphi_2) + A_3 \cdot \cos(3\omega_0 t + \varphi_3) + \dots$$



← Pro jiné než harmonické signály roste přesnost approximace s rostoucím stupněm harmonických funkcí!

- Teoreticky jsou potřeba všechny průběhy z bázového prostoru
- Všimněte si → čím ostřejší hrany → tím pozvolnější pokles spektra

## 2 Převod mezi časovou a frekvenční oblastí (analýza signálu)

$$x(t) \Leftrightarrow X(\omega)$$

- $t$  – časová (prostorová) oblast
- $\omega$  – frekvenční oblast  $(\omega = 2\pi f)$

### Základní definiční vztahy:

přímá spojitá FT:

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-i2\pi f t} dt$$

zpětná spojitá FT:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(f) e^{+i2\pi f t} df$$

Požadavky na  $f(t)$  (ale i  $X(f)$  !):

- alespoň po částech spojitá (integrovatelnost) – výpočet nutno rozložit, ale lze provést!
- integrál s konečnou hodnotou  $\int_{-\infty}^{+\infty} |(x(t))| dt = k; k < \infty$

#### Paresevalova rovnost:

$E = \int_{-\infty}^{\infty} |x(t)|^2 dx = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega$  - vyjadřuje ekvivalenci energií v časovém (prostorovém) a spektrálním prostoru, **musejí se rovnat** (důležité při dalším zpracovaní na *dsp...*)

$X(\omega)$	- <b>komplexní</b> spektrum (pro reálné signály sudé), vhodné pro další výpočty
$ X(\omega) $	- amplitudové spektrum, (pro reálné signály sudé), amplituda jednotlivých harm. složek
$\arg(X(\omega))$	- fázové spektrum (pro reálné signály liché), amplituda jednotlivých harm. složek
$ X(\omega) ^2$	- výkonové spektrum, výkon nesený jednotlivými harmonickými složkami ( $R = 1W$ )

$$|X(\omega)| = \sqrt{\Re(\omega)^2 + \Im(\omega)^2} \quad \phi(\omega) = \text{atan}\left(\frac{\Im(\omega)}{\Re(\omega)}\right)$$

### 3 Jak vypadá spektrum některých periodických funkcí?

Harmonický signál  $x(t) = A \cdot \cos(\omega_0 t)$

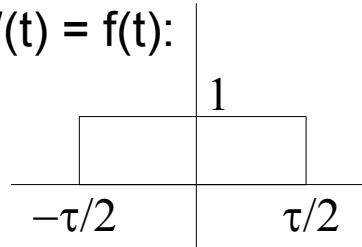
uvažujme omezený signál (vybereme kousek pomocí „výběrového okna“  $w(t)$ )

nejjednodušší  $w(t) =$  

$$W(t) = f(t):$$

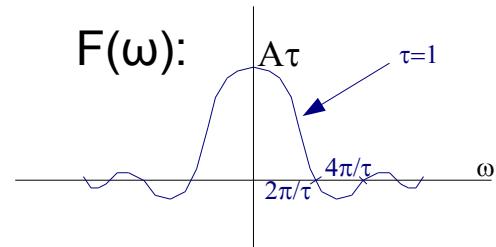
z minula:

Jak spočítat **spektrum obdélníku**?:



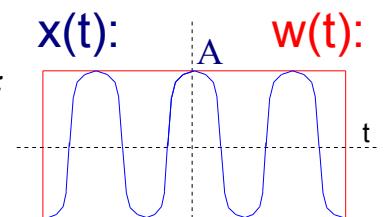
$$F[W(t)] = \int_{-\tau/2}^{\tau/2} A \cdot 1 \cdot e^{-j\omega t} dt = A \cdot \left[ \frac{e^{-j\omega t}}{-j\omega} \right]_{-\tau/2}^{\tau/2} = A \cdot \frac{e^{-j\omega\tau/2} - e^{j\omega\tau/2}}{-j\omega} = A \cdot \frac{+e^{j\omega\tau/2} - e^{-j\omega\tau/2}}{j\omega} \cdot \frac{2}{\omega} = \dots$$

$$\dots = A \cdot \frac{\sin\left(\omega \frac{\tau}{2}\right)}{\omega} \cdot \frac{1}{\frac{\tau}{2}} = \dots = A \cdot \tau \cdot \frac{\sin(x)}{x}$$

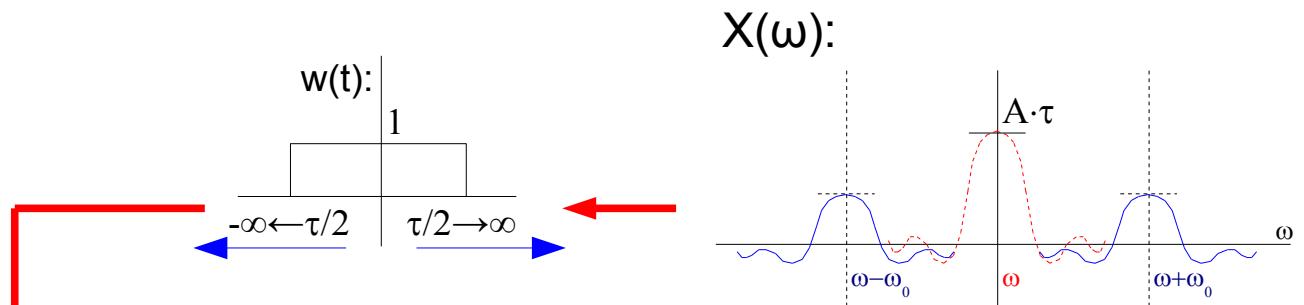


Spektrum „kousku“ harmonického signálu - vybraného / vynásobeného  $w(t)$ :

$$\cos(\omega_0 \cdot t) \cdot w(t) = \frac{e^{j\omega_0 t} + e^{-j\omega_0 t}}{2} \cdot w(t) = \frac{1}{2} \cdot e^{j\omega_0 t} \cdot w(t) + \frac{1}{2} \cdot e^{-j\omega_0 t} \cdot w(t)$$



$$X(\omega) = F[\cos(\omega_0 t) \cdot w(t)] = \frac{1}{2} \cdot A \cdot \tau \cdot \frac{\sin\left((\omega - \omega_0) \cdot \frac{\tau}{2}\right)}{(\omega - \omega_0) \cdot \frac{\tau}{2}} + \frac{1}{2} \cdot A \cdot \tau \cdot \frac{\sin\left((\omega + \omega_0) \cdot \frac{\tau}{2}\right)}{(\omega + \omega_0) \cdot \frac{\tau}{2}}$$



$$x(t) = \sin(2\pi f_0 t)$$

$$\Leftrightarrow$$

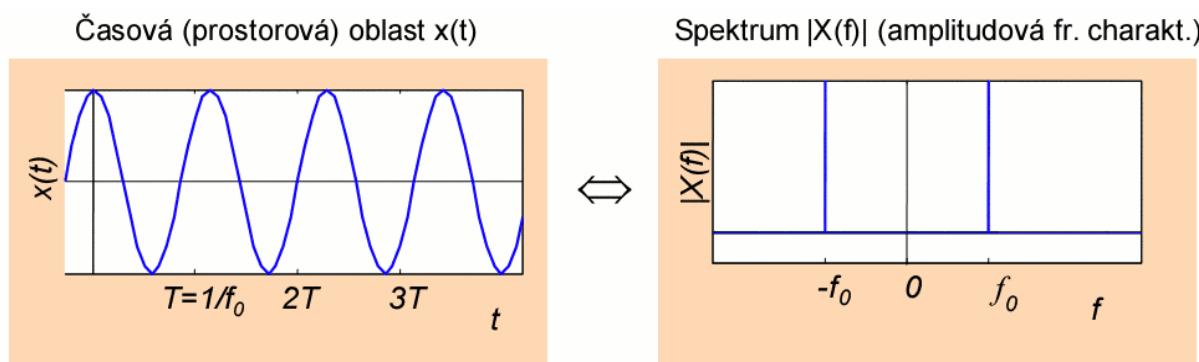
$$X(f) = \frac{1}{2i} (\delta(f - f_0) - \delta(f + f_0))$$

$$\text{Def. FT: } X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} dt$$

$$\text{Eulerovy vzorce: } e^{\pm j2\pi f_0 t} = \cos(2\pi f_0 t) \pm i \sin(2\pi f_0 t)$$

$\delta(f)$  ... Diracova funkce





### 3.1 Diskrétní FT

Diskretizace v čase:

$$t \rightarrow t_n \quad n$$

Diskretizace ve frekvenci:

$$f \rightarrow f_k \quad k$$

## Diskrétní Fourierova transformace

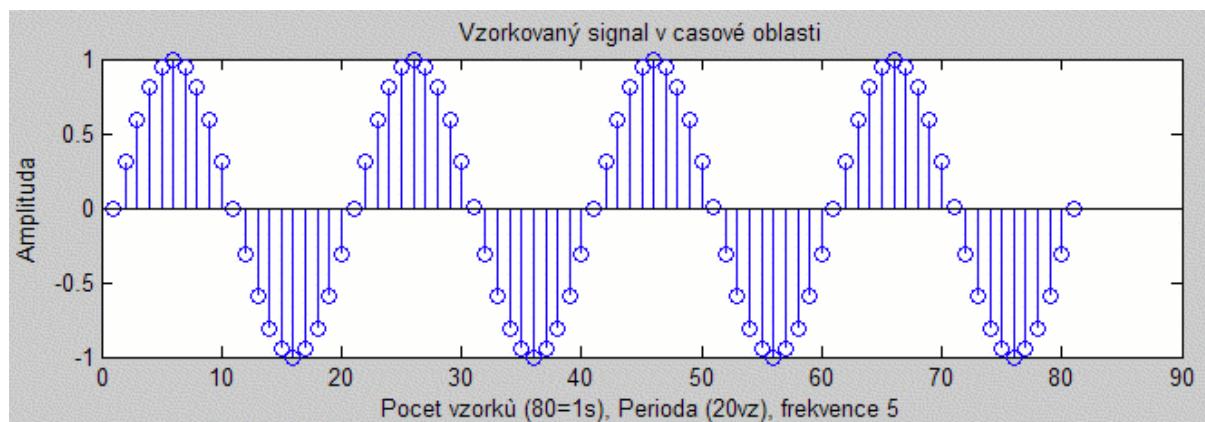
Přímá diskrétní FT:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-i 2\pi \frac{nk}{N}}$$

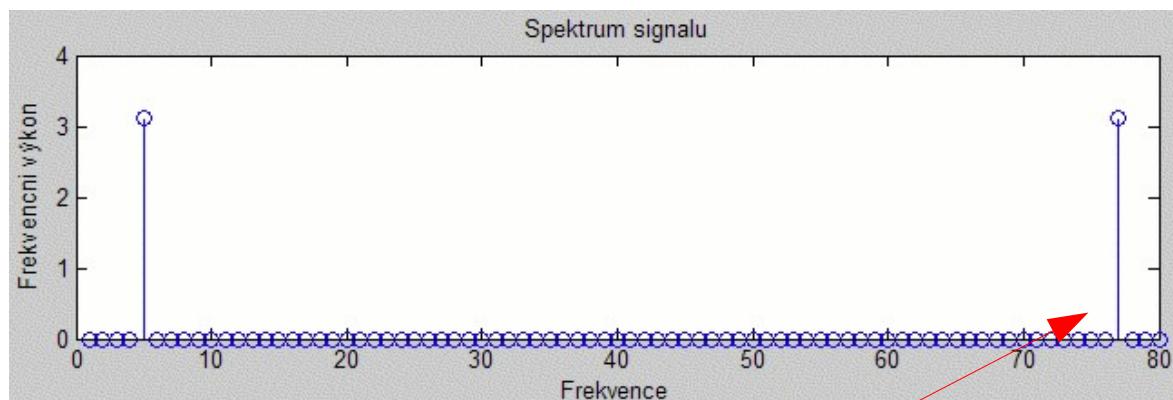
$$x(n) \Leftrightarrow X(k)$$

Zpětná diskrétní FT:

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{+i 2\pi \frac{kn}{N}}$$



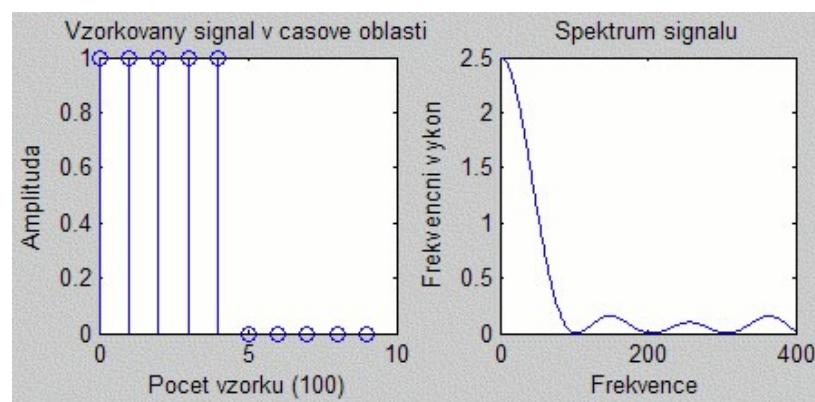
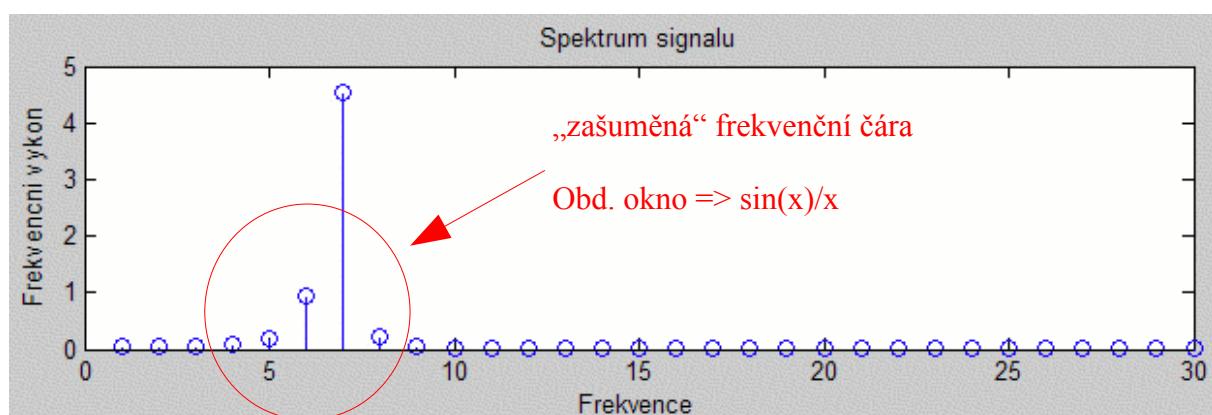
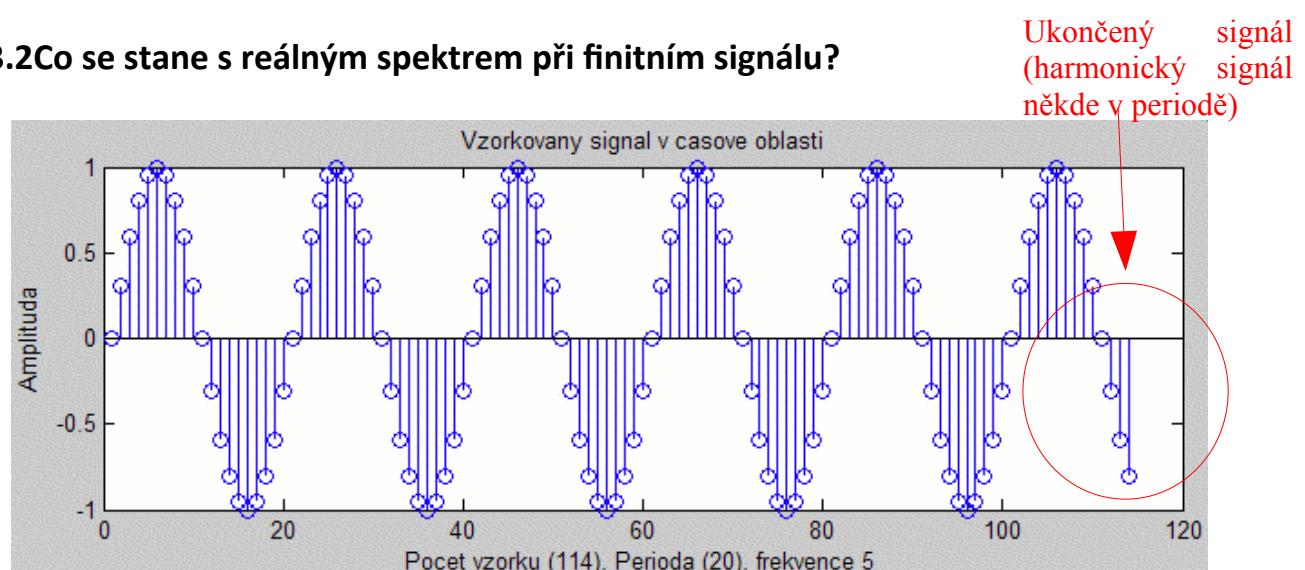
harmonický signál ( $f(t)$ ) „oříznutý“ přesně na konci periody



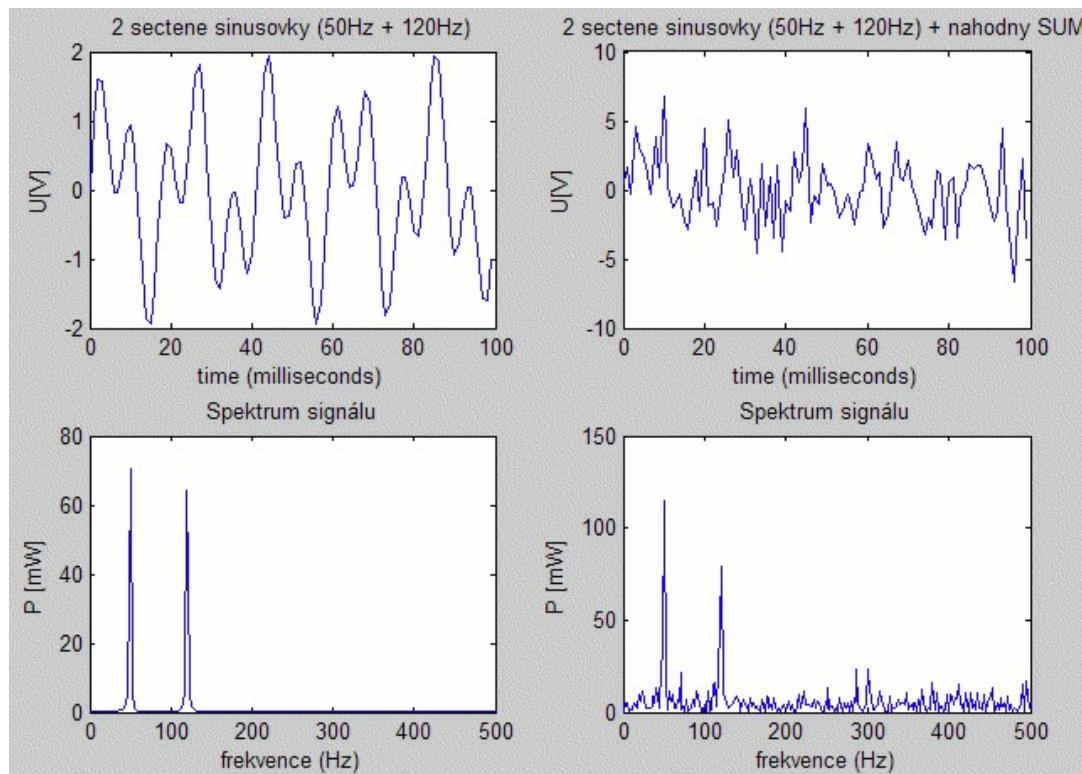
Horizontální osa: frekvence

- $k$  – index (0 – 81)
- $w_k$  – kruhová frekvence (0 - 2  $\pi$ )
- normalizovaná frekvence (0 - 1)

### 3.2 Co se stane s reálným spektrem při finitním signálu?



## Signál ovlivněný šumem



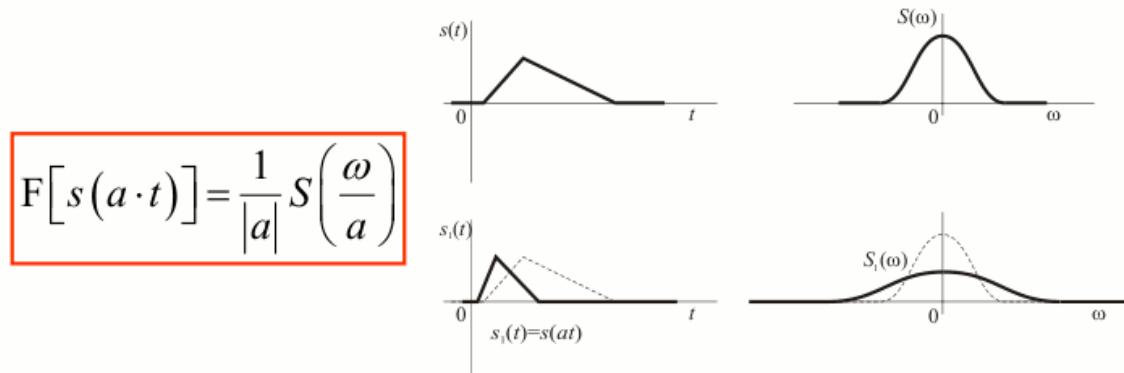
### Jedna z vět FT:

- věta o posunu – posun objektu v předmětovém prostoru vede k fázovému posunu ve spektrální oblasti

$$F[s(t-t_d)] = \int_{-\infty}^{\infty} s(t-t_d) \cdot e^{-j\omega t} dt = \int_{-\infty}^{\infty} s(\tau) \cdot e^{-j\omega\tau+t_d} d\tau = S(\omega) \cdot e^{-j\omega t_d}$$

- Věta o změně časového měřítka

$$F[s(at)] = \int_{-\infty}^{\infty} s(at) e^{-j\omega t} dt = \begin{cases} \frac{1}{a} \int_{-\infty}^{\infty} s(t) e^{-j\omega \frac{t}{a}} dt & \text{pro } a > 0 \\ -\frac{1}{a} \int_{-\infty}^{\infty} s(t) e^{-j\omega \frac{t}{a}} dt & \text{pro } a < 0 \end{cases} = \frac{1}{|a|} \int_{-\infty}^{\infty} s(t) e^{-j\frac{\omega}{a}t} dt = \frac{1}{|a|} S\left(\frac{\omega}{a}\right)$$



$$F[s(at)] = \frac{1}{|a|} S\left(\frac{\omega}{a}\right)$$

- S výhodou možno použít na „zpřesnění“ výpočtu spektra prodloužením nulami – viz help: [fft](#)

## 4 Dvoudimenziornálí FT

Přímá 2D FT

$$F(u, v) = \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} f(x, y) e^{-j2\pi(\frac{ux}{X} + \frac{vy}{Y})}$$

Zpětná 2D FT

$$f(x, y) = \frac{1}{UV} \sum_{u=0}^{U-1} \sum_{v=0}^{V-1} F(u, v) e^{+j2\pi(\frac{ux}{U} + \frac{vy}{V})}$$

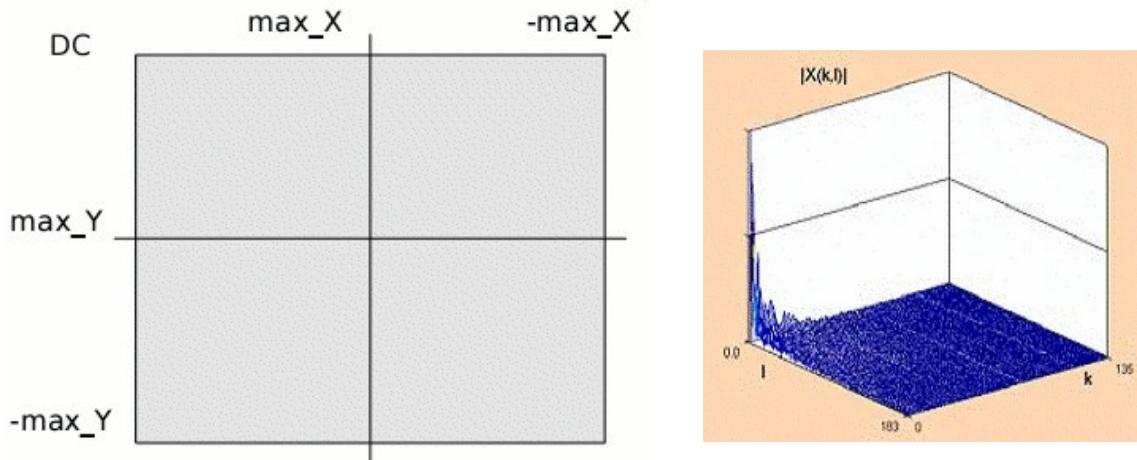
**2D FT** je matematicky upravitelná (separovatelná) tak, že může být nejprve vypočtena **1D FT** všech řádků, následovaná **1D FT** všech sloupců:

$$F(u, v) = \sum_{x=0}^{X-1} \left[ \sum_{y=0}^{Y-1} f(x, y) e^{-j\frac{2\pi ux}{X}} \right] e^{-j\frac{2\pi vy}{Y}}$$

tedy 2D FT má poměrně jednoduché rozšíření do dvou stavů:

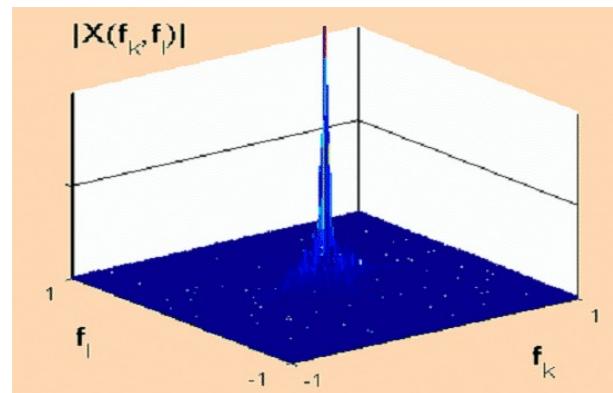
- zjednodušeně: provede nejprve FT sloupců a pak FT řádek

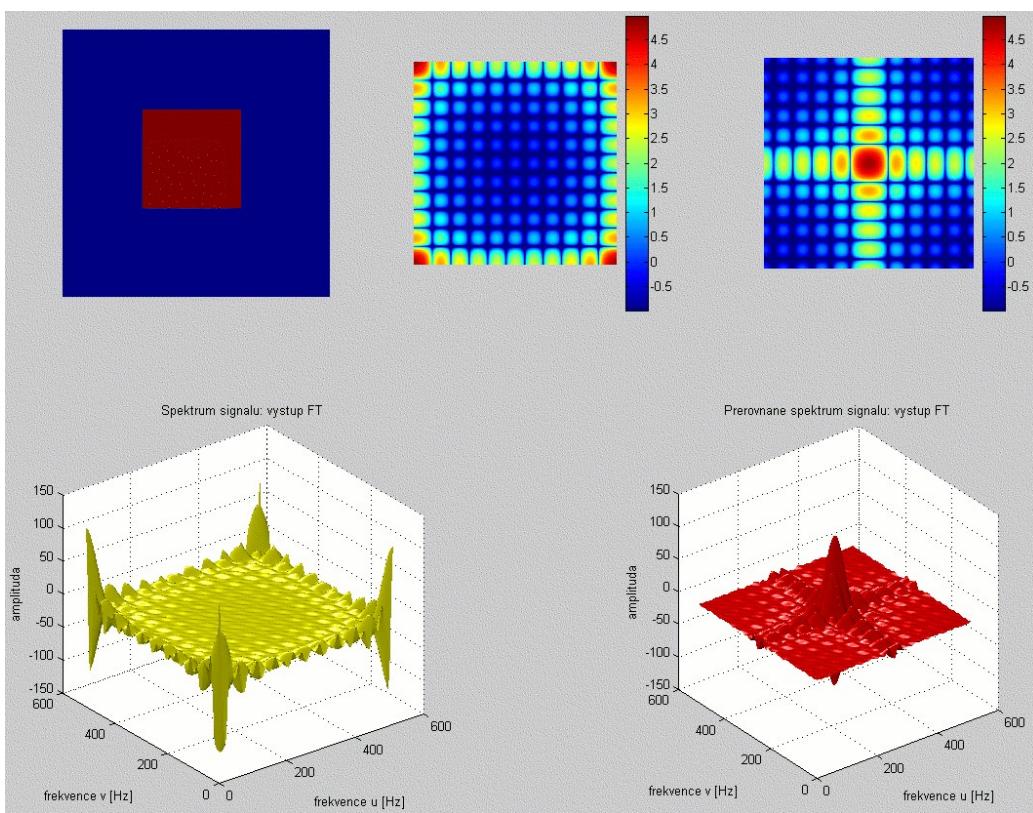
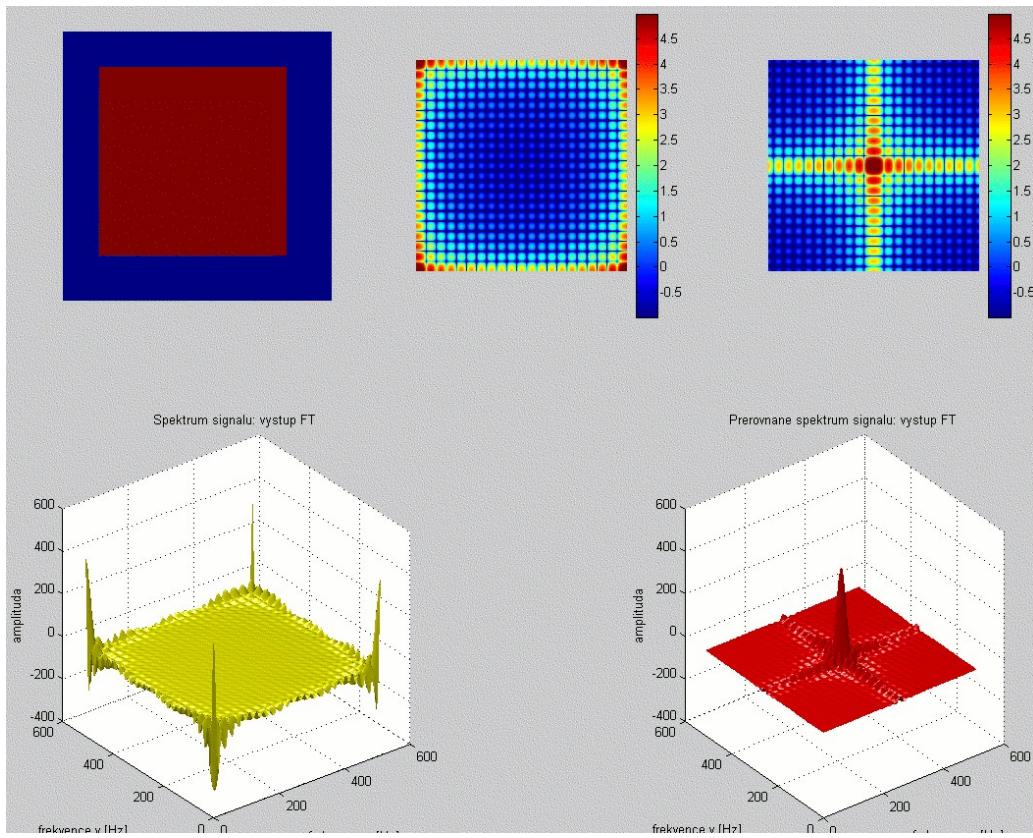
**Vizualizace 2D FT spektra:** - spektrum navzorkovaného signálu je periodické!!!

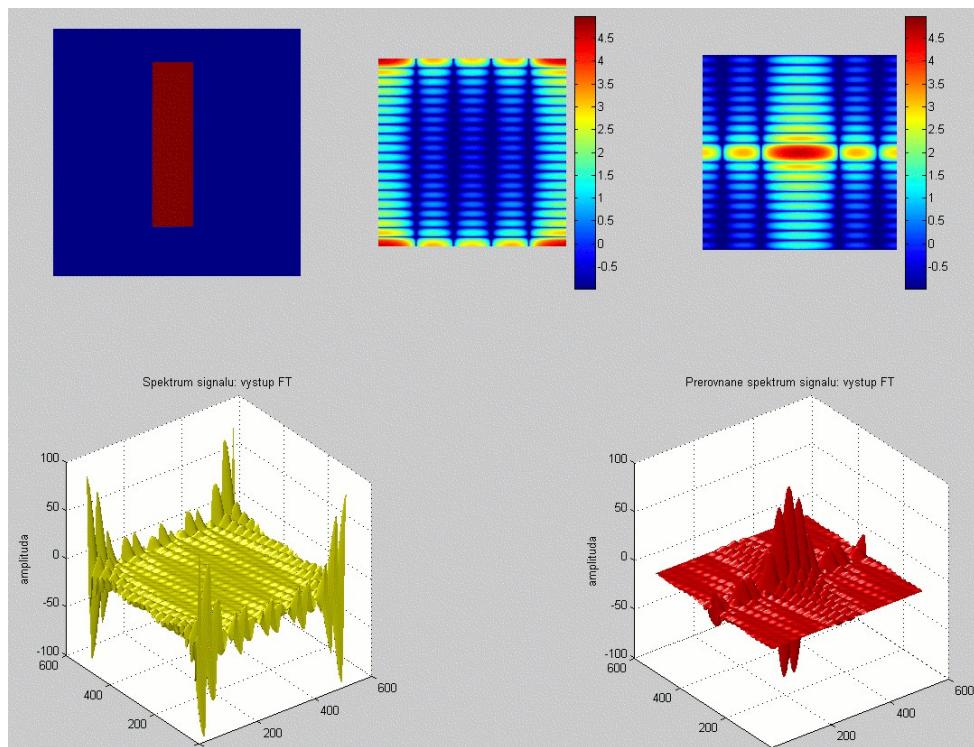


Tedy DC se nachází na „krajích“ spektra

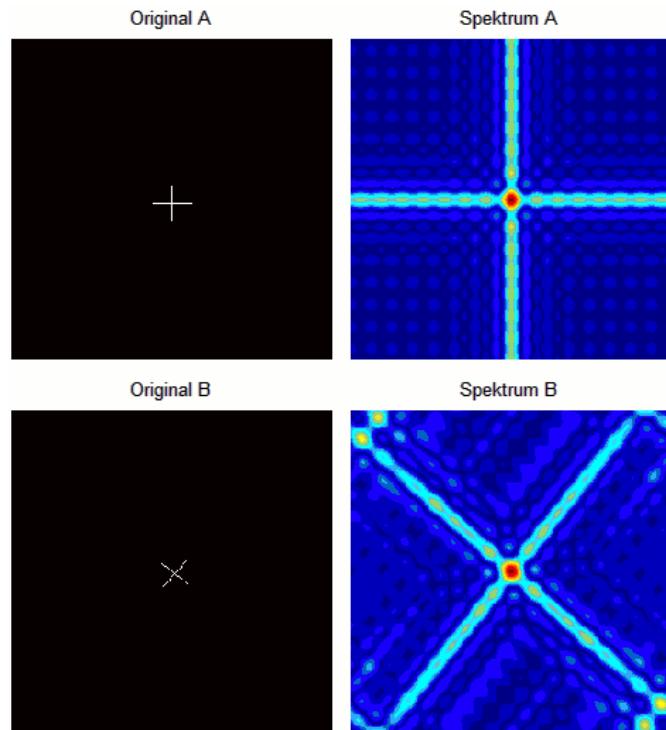
- pro intuitivnější práci přeskupujeme kvadranty do „srozumitelnější podoby“ s DC uprostřed



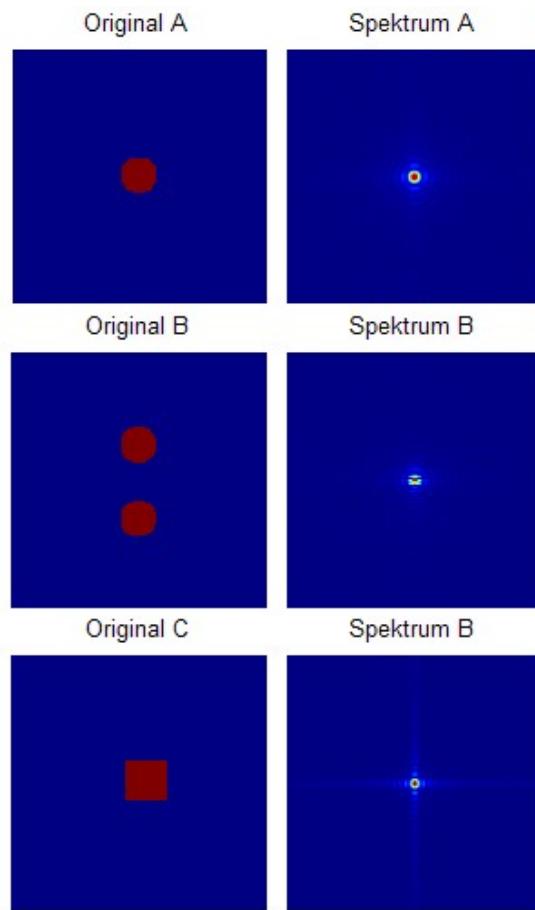




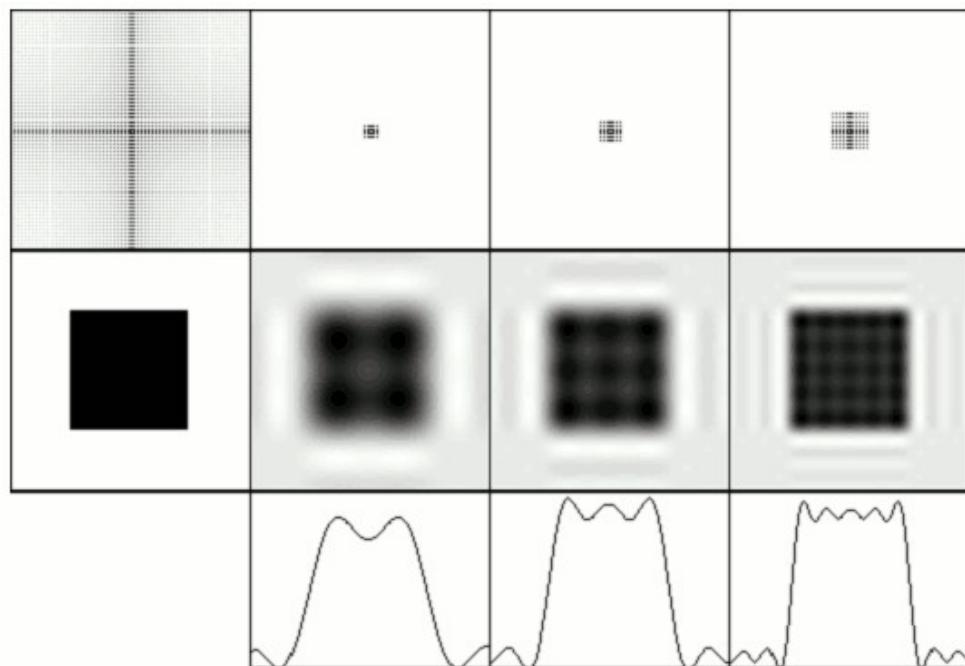
Otočení předlohy → otočení spektra:



Spektrum pulzního průběhu



### Oříznutí části spektrálních koeficientů



# Principy snímání obrazové a zvukové informace, principy snímacích prvků.

## Obsah

1 Lidské vidění.....	1
2 Analogové snímače (digitální snímače s analogovými výstupy).....	3
2.1 Filmová technika rozkladu obrazu:.....	3
2.2 Videosignál (CCIR):.....	3
3 Nábojově vázané prvky (Charge Coupled Devices — CCD).....	5
4 Barevné snímání.....	9
5 Systémy pro snímání nízkých intenzit osvětlení.....	14
5.1 Princip funkce.....	14

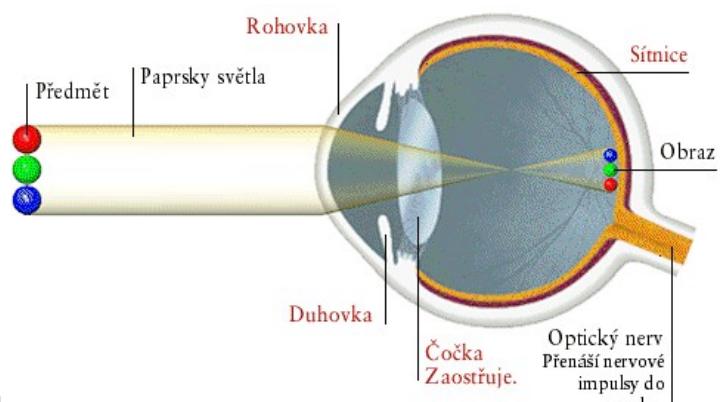
Snímání obrazové / zvukové inf.:

- *převod jednoho fyzikálního vyjádření na jiné* (el. mag / akustické vlnění na el. signál) – optoelktrická transformace
  - při digitalizaci – zároveň převod do číselného vyjádření
- konečným článkem řetězu většinou lidské oko – optimalizace, zjednodušení, zanedbání

## 1 Lidské vidění<sup>1</sup>

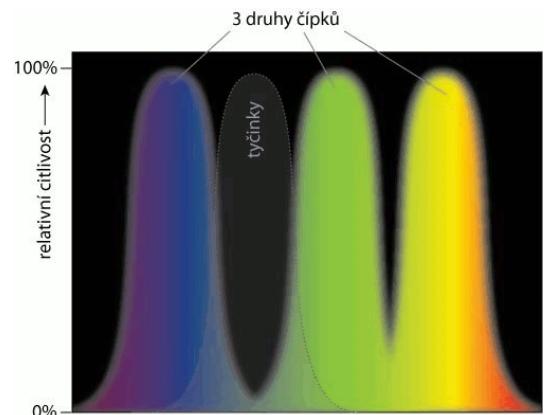
Lidské oko:

- transf. elmg. záření → chemická reakce → el. potenciál
- >100mil fotoreceptorů, ale pouze 1.6M nervových vláken do mozku (slepá skvrna)
- duhovka s otvorem uprostřed = pupila (obdoba AVC)
  - při silném osvětlení – svalová redukce - omezení průch. světla
  - v epitelu duhovky barevný pigment (nejméně pigmentu = modrá)
- čočka – cca 60D (1/ohnisk. vzd), ohnisková vzdálenost ≈ konstantrní



<sup>1</sup> <http://webvision.med.utah.edu/>

- umožňuje [svalovou akomodaci](#) (omezená akom. - omezené možn. ostření)
- zmenšený obrácený obraz - postprocessing
- fotoreceptor – až 80% všech inf.
  - tyčinky ( $>100\text{Mil}$ ) – citlivé na malou int. světla, č/b vidění
  - čípky (3-6Mil) – méně citlivé, aktivují se jen při vyšší int. sv.
    - největší konc. v tzv. žluté skvrně, ostření = zaměření do žl. sk
    - trojího druhu *RGB* – nejvyšší „spektrální“ koncentrace mezi modrou a zelenou (genetický vývoj savců – zelené okolí):
  - $>100\text{Mil} \neq >100\text{MPix}$ 
    - fotoreceptory nepřenášejí inf. do mozku přímo - počet nervových vláken mnohem nižší – mezistupeň zpracování na úrovni sítnice



Rozsah vnímání lidského oka 390 - 760 nm (770 - 395 Thz,  $E = 3,18 - 1,63 \text{ eV}$ )

- maximum vyzařování slunce kolem 500nm (dáno teplotou povrchu)
- atmosféra v této oblasti minimálně pohlcuje záření

Proč nevidíme více → UV záření?

- nižší  $I$  → vyšší  $f$  → vyšší  $E$  → vyšší rozklad rhodopsinu, poškození fotoreceptorů
- biologický materiál (%voda) → nevhodný index lomu pro stavbu čočky pro UV záření
- hmyz → jiná stavba oka - složené oko bez čoček → částečně vnímá UV

Proč nevidíme → IR záření?

- v oblasti IR přirozeně září okolí, lidské tělo → tkáň v oku → vlastní šum
- více se projevuje difrakce, ohyb paprsků → rozmažání

Zjednodušený princip vidění<sup>2</sup>:

- foton <sup>$E$</sup>  → rhodopsin => opsin<sup>+</sup> + retinal<sup>-</sup> = el. potenciál (pouze po dobu pár ns)
  - příliš velká Int. → kompletní rozklad rhodopsinu → krátk. / nevratné oslepnutí
- syntéza: opsin<sup>+</sup> + retinal<sup>-</sup> + vit. A = obnovený rhodopsin
  - nedostatek vit. A - šeroslepota

<sup>2</sup> Obecný princip vidění znám přibližně 350 let. Do té doby naivní představy – Pythagoras (- 500) – oko „ohmatává“ okolí pomocí emise paprsků..., přesto komplikované operace oka, podrobně známo 100 let

- ve tmě menší rozklad rhodopsin → po 10 – 20 min zvýšená konc. rhodopsinu → daleko vyšší citlivost oka (adaptace na tmu)
- společná vlastnost – **setrvačnost** (doba chem. reakce) – využíváno tech. prostř.

## 2 Analogové snímače (digitální snímače s analogovými výstupy)

Základním principem analogového snímání je *rozklad* obrazu do elementárních elementů.

- řádků / snímků
  - původně elektro-mechanický princip
- stále využíván při přenosu / u snímacích kamer
- snímání prováděly: snímací elektronky



DÚ: shlédnout: *Vratislav Rýpar - Počátky televizní techniky*

<https://www.youtube.com/watch?v=L8IcmpeyUrg>

### 2.1 Filmová technika rozkladu obrazu:

lidské oko cca >20Hz → pohyb (lokální regenerace rhodopsinu)

filmový pás 24Hz, oko vnímá jako pohyb (fotoreceptory v sítnici, rychlejší tyčinky - jas),

nízké frekvence změn jasových hodnot – oko vnímá jako blikání →  $2 \times f_p$  (s uzávěrkou)

### 2.2 Videosignál (CCIR<sup>3</sup>):

počet snímků za 1s = **25** → i zde nízké fr. v obrazu způsobují blikání → *prokládání*

→ přenos jednoho snímku 40ms

→ prokládání: 1. *půlsnímek* začíná na 1. řádku, 2. půlsnímek začíná na 313 ř.

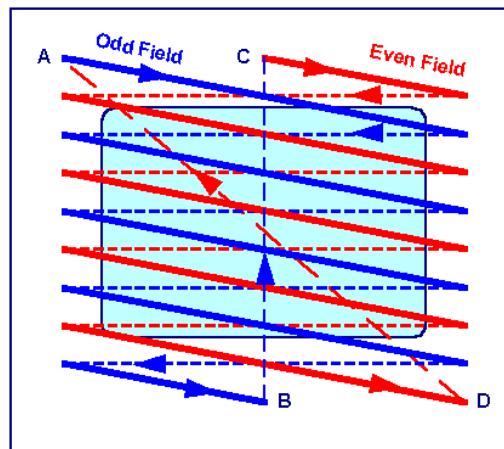
jeden snímek = **625** (525) řádků

horizontální frekvence (1 / doba jednoho ř.):  $1 / 25 / 625 = 64\text{ms} = \textcolor{blue}{15625\ Hz}$

*vychylovací napětí* → *vychylovací cívky*

---

<sup>3</sup> Comité consultatif international pour la radio – mezinárodní poradní radiový výbor, součást ITU - International Telecommunication Union – mezinárodní telekomunikační unie



Obrázek 1: Princip prokládání

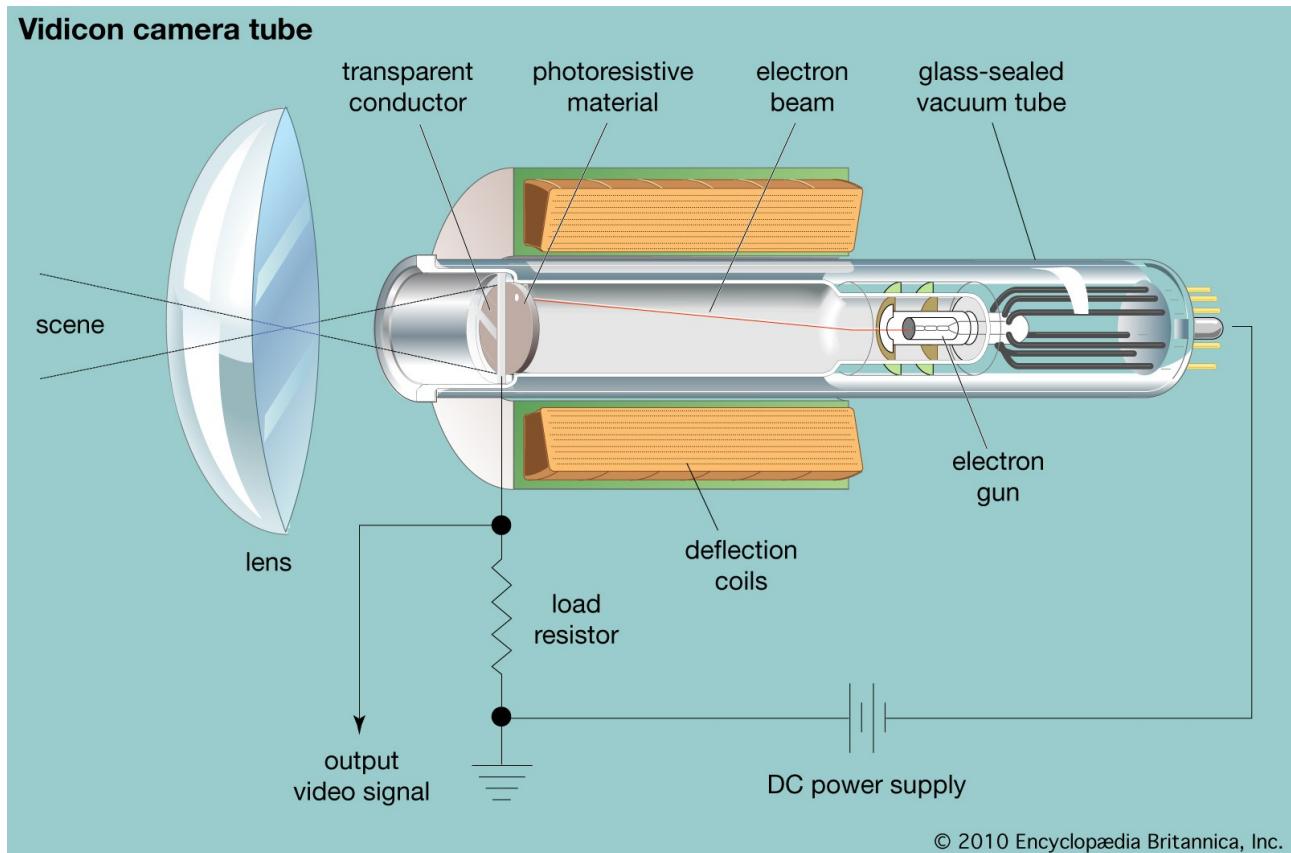
počet bodů na ř.: cca 800

$$\rightarrow 800 \text{ b/ř} \times 625 \text{ ř} \times 25 \text{ s}^{-1} \approx 13 \text{ MS}$$

v krajním případě se střídá č. s b. (jemná šachovnice)

≈ approximace **6.5 MHz** sinusovkou (min cca: 50Hz)

*modulační napětí → katoda obr.* ( $U_{white} < U_{dark}$ )

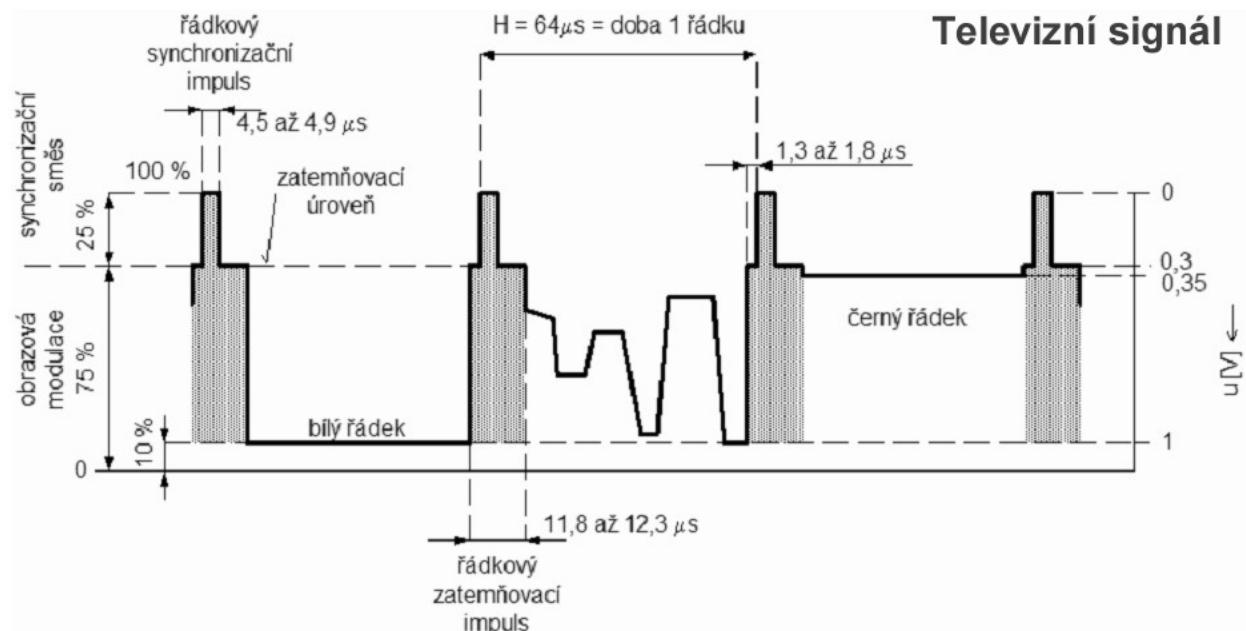


© 2010 Encyclopædia Britannica, Inc.

Obrázek 2: Snímací elektronka - Vidikon

Rozlišení obrazu:

- 625 (CCIR) řádků  $\times 4/3 \approx 800$  bodů / řádek
  - ne všechny viditelné - cca 82%, teletext, zat. pulz...)
  - rozlišení TV PAL CCIR 576 x 768  $\approx 0.5\text{Mpix}$ .



Obrázek 3: Videosignál CCIR

**3 Nábojově vázané prvky (Charge Coupled Devices – CCD)<sup>4</sup>**

**Vznik:** při vývoji polovodičových pamětí – mají stejnou architekturu.

**Funkce:**

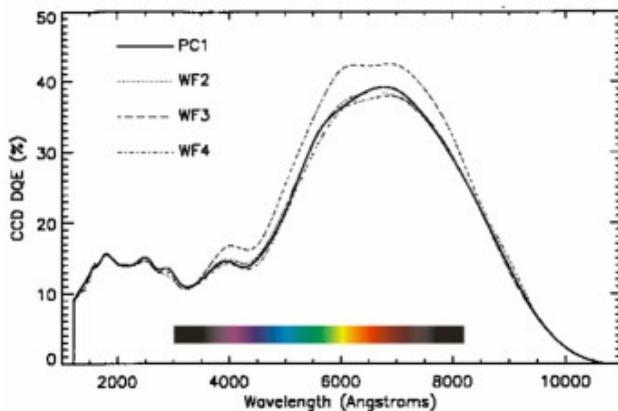
- přicházející světlo (fotony) vytváří v polovodiči elektrický náboj (uvolněné elektrony).
- Akumulované náboj. pakety  $\rightarrow$  úměrné světelné exp.  $E$  a době akumulace  $t_a \rightarrow$  přesun na výstupní převodník  $Q \rightarrow U_o$  (přes analogový posuvný registr)



- úměrné s tzv.  $QE$  - quantum efficiency ( $>80\%$ )  $QE = \frac{el./s}{fot./s}$

---

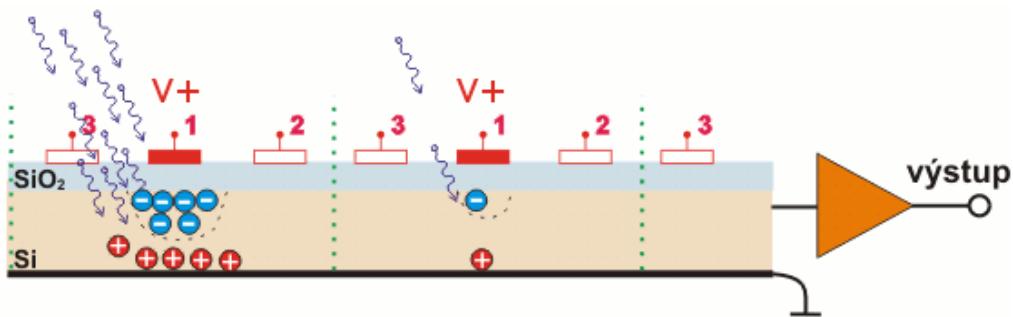
<sup>4</sup> 1969 – Bell Laboratories



Obr. 1: průběh Quantum efficiency v závislosti na  $\lambda$  pro CCD snímače Hubbleova teleskopu (120 nm - 1100 nm)

### Princip:

- elektrony se *nemohou volně pohybovat* po čipu - na čipu jsou vytvořeny *svislé a vodorovné negativní potenciálové valy* (odpuzující elektrony) - mřížka tzv.:
  - „potenciálových studní“ - vytvořeny z *elementárních kapacitorů* (MIS<sup>5</sup>), z nich elektrony nemohou uniknout.



- pixely vystavené většímu množství světla *nakumuluji* více elektronů

### Vstupní obrazový signál

(pouze pro inf.! )



- je v čase vzorkován a pro velikost jednoho vzorku platí:

$$U_{OI}(\lambda) = t_a \int_{\lambda_{min}}^{\lambda_{max}} S(\lambda) \cdot E(\lambda) d\lambda + U_r \cdot t_a$$

- kde:
  - $t_a$  - doba akumulace náboje v potenciálových jamách (např. pro PAL typ. 40 ms)
  - $S(\lambda)$  - spektrální citlivost snímače [ $A \cdot J^{-1}m^2$ , případně  $V \cdot lx^{-1} \cdot s^{-1}$ ].
  - $E(\lambda)$  - spektrální hustota záření (intensita osvětlení)  $i$ -tého bodu [ $W \cdot m^{-2}$ , příp.  $lx$ ].

<sup>5</sup> Metal Insulator Semiconductor

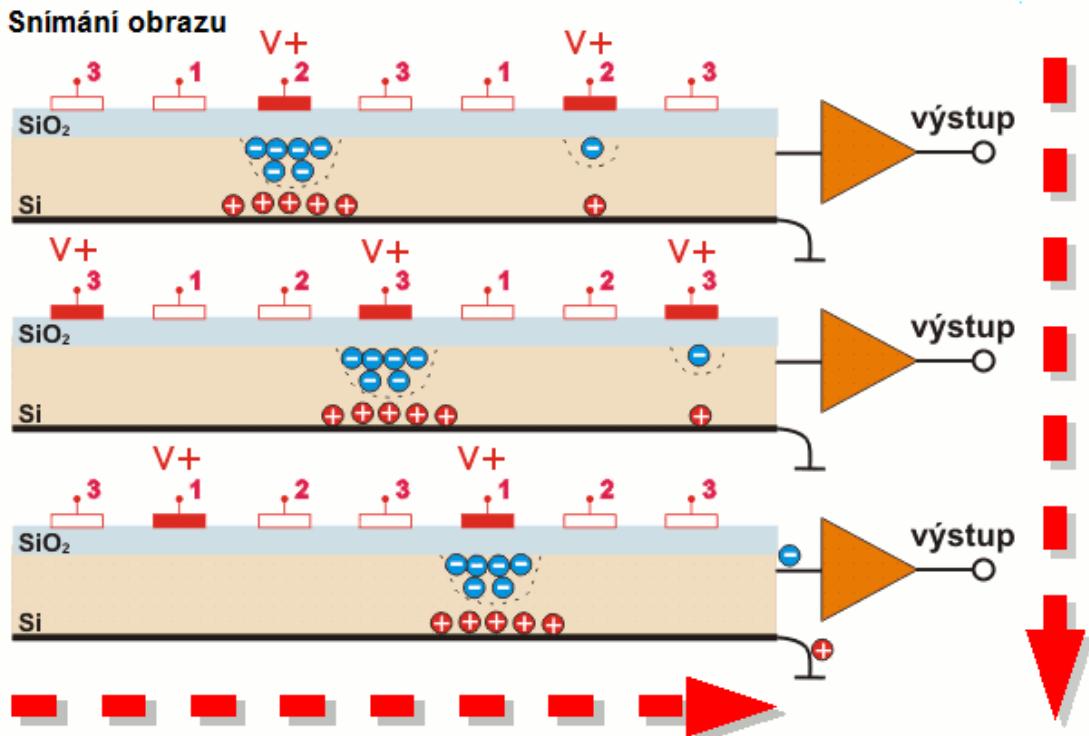
- $U_r$  - souhrnný rušivý signál, tvořený především zbytkovým signálem v černé (*Dark Signal*)  $U_{DS}$  a šumovými signály  $U_s$ .
- každá potenciálová studna vytváří reprezentuje jeden obrazový bod – pixel
  - pixel - picture element
- omezený rozsah intenzit (malá C pixelu) – *brzy nastává saturace – pomalá regenerace!*

#### *Vlastnosti:*

- odezva na intenzitu osvětlení je lineární
- mohou poskytovat přím digitální obraz
  - často jsou ve spojení s AD-DA převodníky i ve snímačích s analogovým výstupem
- mají schopnost akumulace náboje
  - mohou postupně nashromáždit dostatek světla i z velmi slabých světelných zdrojů
- malé rozměry, minimální geometrické zkreslení, odolnost proti magnetickým polím, nízký příkon, velký dynamický rozsah

#### *Vyčítání informace:*

- modulací napětí na elektrodách -> přelévání náboje (analogový posuvný registr)
- ve spodní části „horizontální registr“ -> A/D převodník + zesilovač
  - vyčítání obrazu u CCD = posouvání obrazu



### Organizace:

- řádková struktura - lineární snímače (jednořádkové kamery)
  - technické snímače, skenery, satelitní snímání...
- plošná struktura - obdélníkové snímače
  - kamery, fotoaparáty,

### Výhody (zejména ve srov. s katod. snímači)

- malá velikost
- nízký příkon
- minimální **geometrické zkreslení**
- široký spektrální rozsah
- odolnost proti mg. polím

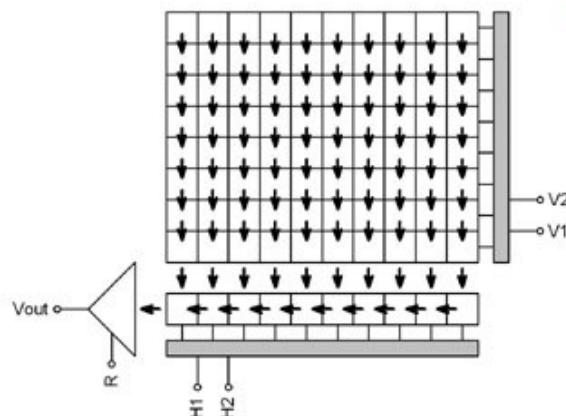
### Nevýhody

- vlastní šum (citlivý na teplotu)
- přelévání náboje (nedokonalá izolace)

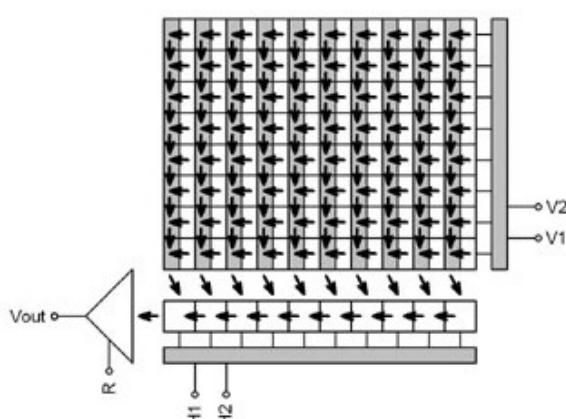
### Typy CCD:

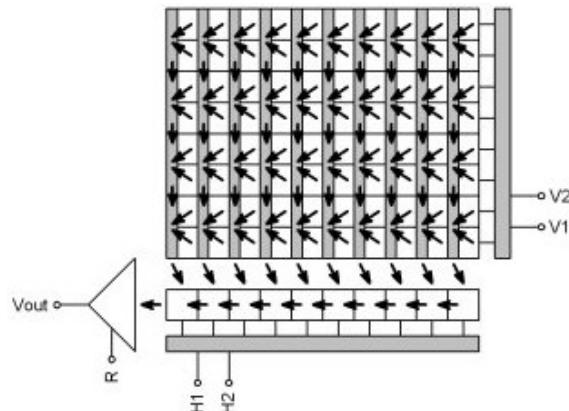
- **FullFrame (FF)** – vystavují celou plochu světlu

- během čtení je potřeba používat mechanickou uzávěrku
- velmi citlivé – vhodné pro snímání slabých zdrojů světla, vysoká QE

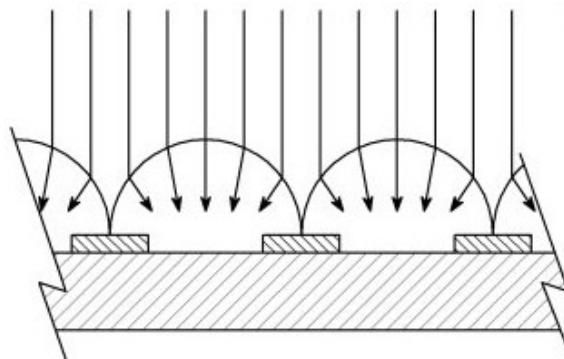


- *Frame Transfer (FT)* – 2 oblasti
  - Imaging area x Storage area (zakrytá oblast)
  - po zachycení obrazu AS  $\rightarrow$  SA a pomalé vyčítání z SA – *elektronická uzávěrka*
  - SA – většinou spodní vrstva – u intenzivních zdrojů hrozí „protečení“ náboje
- *Interline Transfer (IT)* – podobná funkce, jako u FT (*elektronická uzávěrka*)
  - SA je prokládána s IA (např. sudé sloupce mají neprůhlednou vrstvu)
    - při stejné velikosti čipu menší AS – nižší citlivost



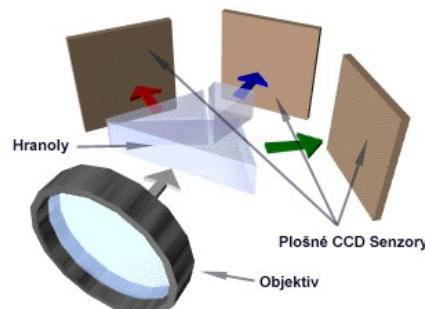


- Zvýšení QE
- mikročočky

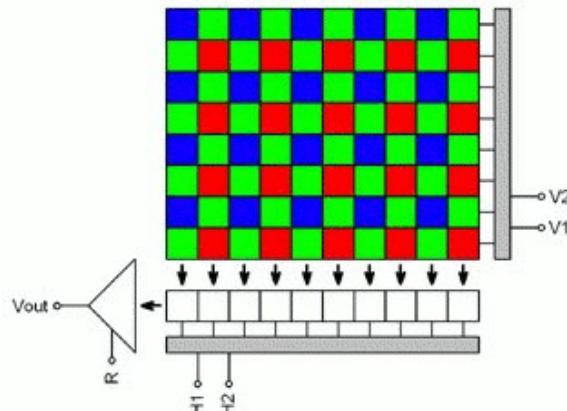


#### 4 Barevné snímání

- aplikace barevných filtrů „na celé čipy“



- aplikace barevných filtrů přímo na pixely



- lidské oko je mnohem citlivější na změny jasu, než změny barev
  - maska zachovává jasové rozlišení
  - $U_{zelené} / 2$  – pouze z důvodu výplně  $2 \times 2$
- barevný filtr RGB – propouští jen omezené množství světla (1/3)
  - jedno z řešení: doplňkový (inverzní) CMY filtr (2/3)
    - problém s výrobou filtru: např. M (= red + not(green) + blue)
      - propouští červ., blokuje zelenou, prop. modrou – kombinace skel, obtížná výroba pro každý pixel

R	G	R	G		
G	B	G	B		
R	G	R	G		
G	B	G	B		

Aditivní barevný model.

C	Y	C	Y		
M	G	M	G		
C	Y	C	Y		
M	G	M	G		

Subtraktivní barevný model.

Barevné CCD snímače mají menší citlivost oproti černobílým snímačům.

pomocí CB snímače je možné vytvořit barevný obraz pomocí filtrů (astr. snímače)

### Nevýhody CCD – vlastní šum

- volné elektrony jsou v čipech uvolňovány i prostřednictvím energ. záření o jiných vlnových délkách
  - tepelné záření o nízkých fr. - způsobuje **tepelný šum (temný proud)**

- udává se v počtu el. za 1s na  $^{\circ}\text{C}$ , např.:  $1.3\text{e}^{-}/\text{s}/^{\circ}\text{C}$

- je možné snížit podchlazením čipu

(u současných kamer se množství E šumu zdvojnásobuje každých cca.  $6-7^{\circ}\text{C}$ )

tepelný šum má při konst. podmínkách konst. momentové charakteristiky

- při uzavřené uzávěrce je možné získat pouze obraz „temného proudu“ - temný snímek, *bias frame*
- temný proud má charakter bílého šumu – technikami sčítání obrazu je možné zvýšit odstup sig. / šum

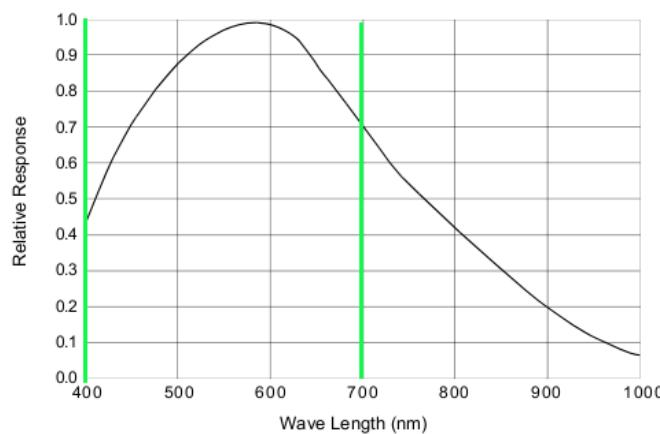
„Širokopásmovou“ závislost je možné využít (viz dále):

Např. většina CCD snímačů „vidí“ velmi dobře i infrač. záření

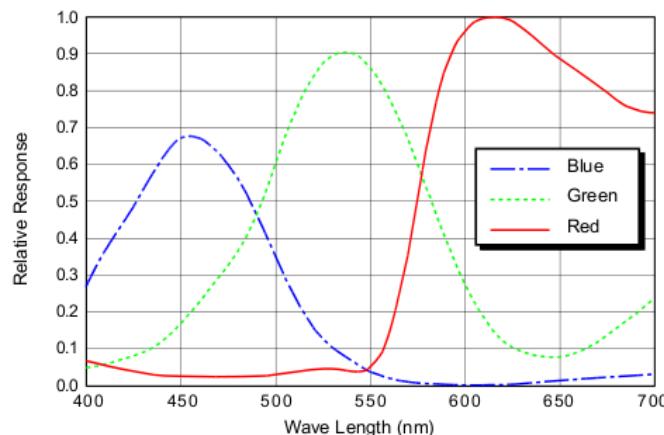
- *č/b snímače jsou v infr. č. spektru mnohem citlivější než bar. CCD*

#### *Odbočka:*

- kvalitní snímač (lab. ZO) – vhodný pro vaše bastlení BP/DiP
- Basler scA1300-32gc
  - 1280x960/32fps
  - propojení – GigEthernet – snadné propojení s dev. boardy
  - 12bit output (linear / LuT)

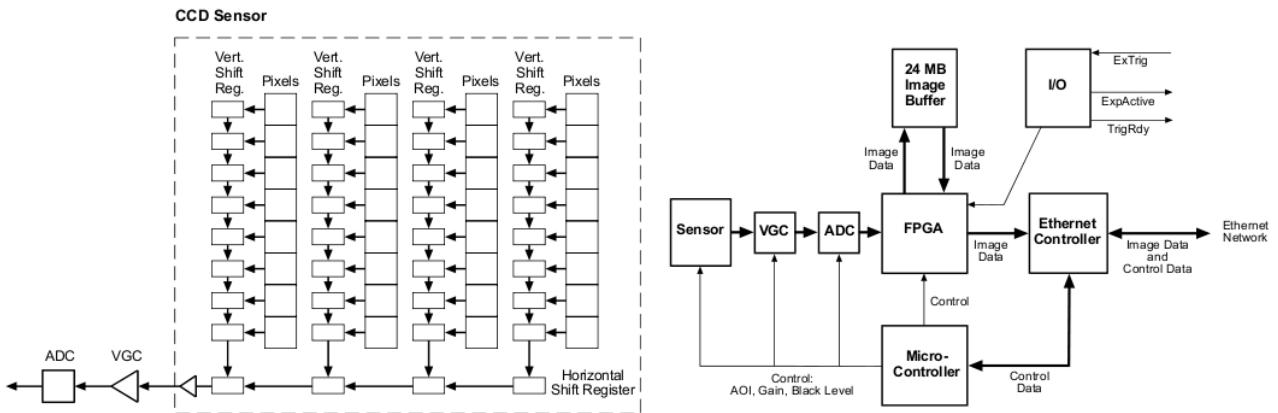


Obr. 2: Spektrální citlivost (souhrnná)



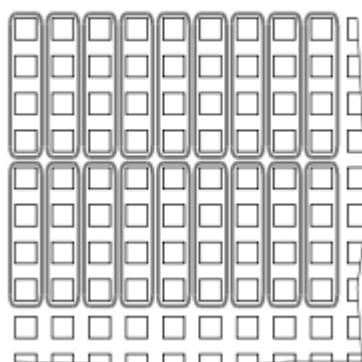
Obr. 3: Spektrální citlivost (jednotlivé snímače)

Zapojení:

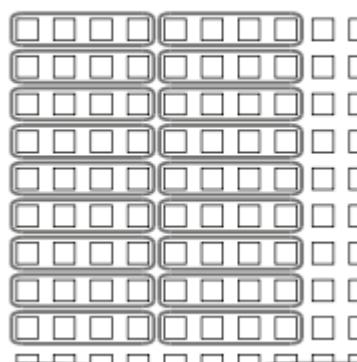


- vnější trigger
- vnější nastavení citlivosti / AVC
- možnost zadat AOI – vyšší fps
- vertical/horizontal Binning 2 až 4

Vertical Binning by 4



Horizontal Binning by 4



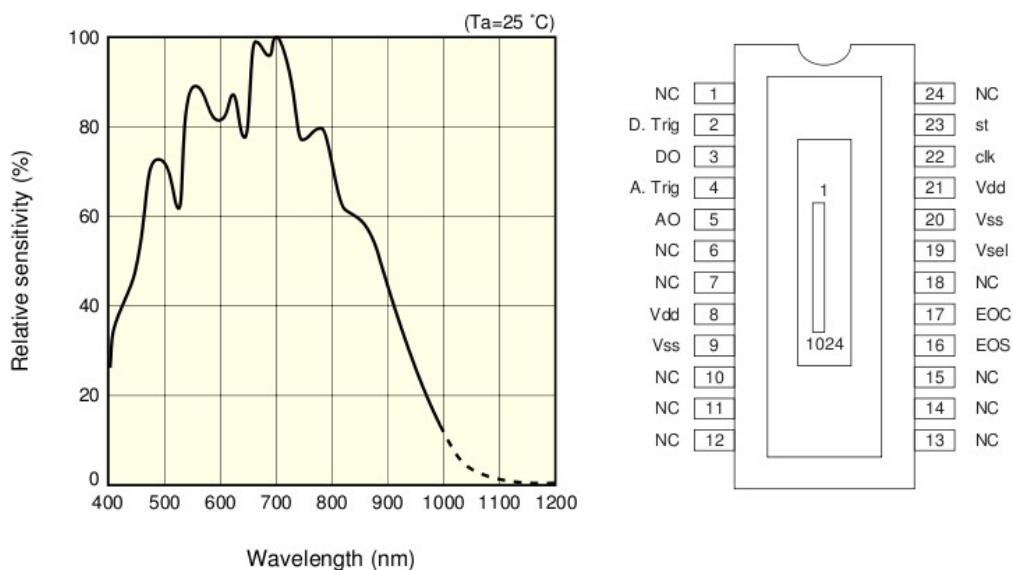
- + mnoho dalších fičurek (hw mirror, hw gama korekce)... hezká hračka

### *Lineární senzor:*

- HAMAMATSU - S10077



- 1 x 1024 px / 10 bit
- DataRate 1Ms/s  $\approx$  1 kHz fps



### *CMOS – snímače*

základní nevýhoda *CCD* – postupné vyčítání inf.

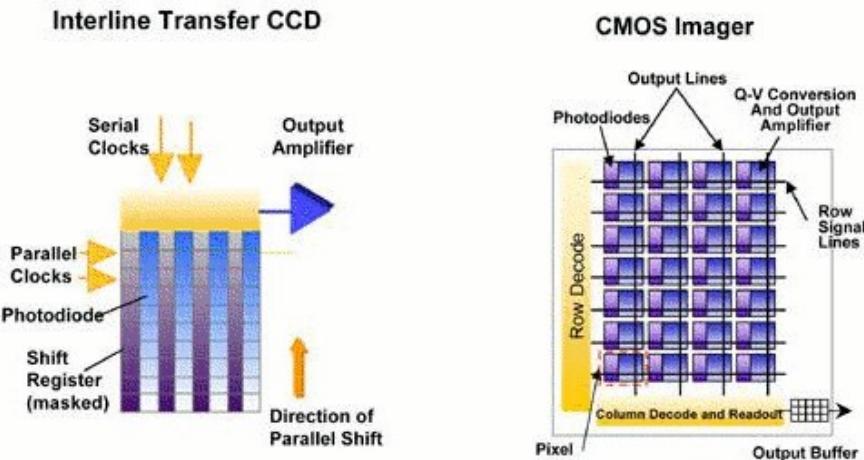
- dále - vysoká cena

*CMOS* každá buňka má vyčítací obvody přímo u sebe – čtení stejně jako *RAM*

- vyráběny stejnou technologií jako mCPU, paměti... => levné
- vyčítání je potom možné „najednou“
- vyčítání trvá kratší dobu – méně energeticky náročné

Výhoda – stejně jako u paměti možnost vyčítání lib. prvku / podle lib. algoritmu

- např. dig. fotoaparáty – nižší rozlišení / rychlejší vyčítání → vyšší *fps*



- Perspektivní technologie, zatím ne tak zvládnutá jako CCD
- nižší citlivost oproti CCD

## 5 Systémy pro snímání nízkých intenzit osvětlení

### 5.1 Princip funkce

#### Běžné optické snímače

- snímají odražené světlo (fotony emitované externími zdroji) – osvětlení sluncem...
- nebo objekty emitující fotony ve viditelné oblasti záření – světla...

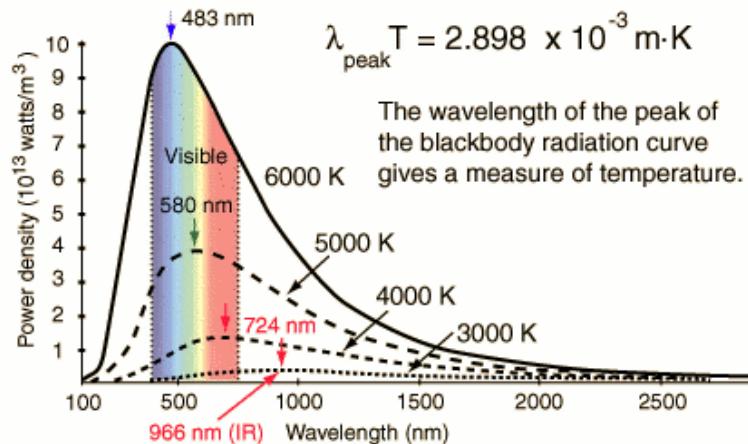
*Tma* – nedostatečné záření ve viditelném spektru – je nutné zvýšit citlivost snímačů

- problém – citlivost např. CCD prvků nelze zvyšovat do nekonečna (vlastní šum...)

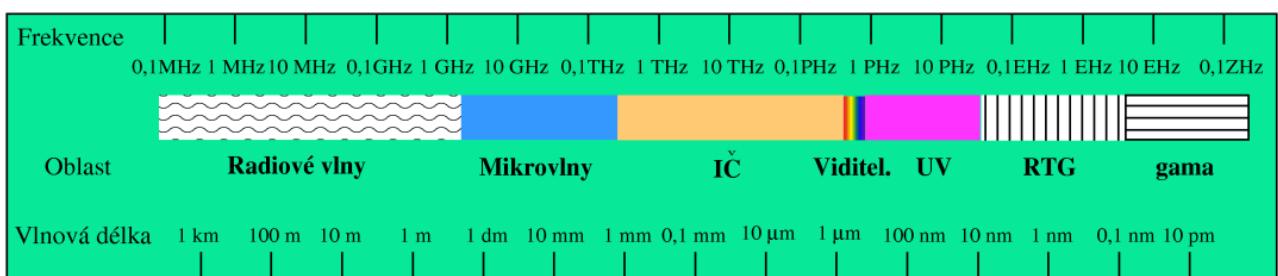
#### Řešení: - snímání jiného záření (*odlišná část spektra*)

- každý objekt s  $T > T_0$  (-273K) emituje záření
  - záření souvisí s neuspořádaným pohybem el. nabitych častic v elektronových obalech atomů
  - záření je absorbováno a emitováno elektrony v atomech, když přecházejí mezi energetickými hladinami
- čím vyšší  $T$  – tím vyšší frekvence *maxima záření* (omezený rozsah – omezené množství elektr. stavů v atomech)

- při teplotách nad  $\approx 525^{\circ}\text{C}$  se  $\lambda < 700\text{nm}$  – viditelné (červené) záření

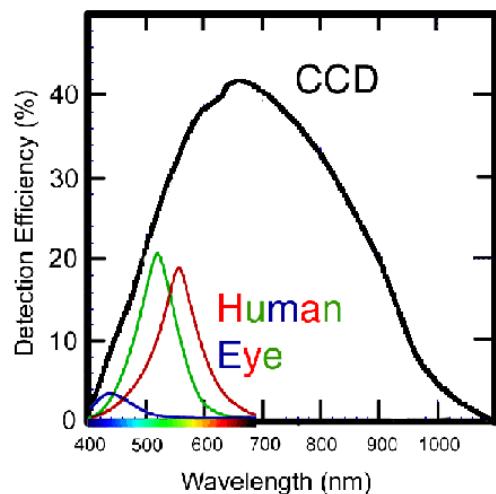


- podle Plancka  $E=hf$ 
  - čím vyšší  $f$  – tím vyšší  $E$  dopadajících fotonů



### Problém

- $E$  fotonů reprezentujících tepelné záření je velmi malá
- nutno zesilovat
  - citlivý snímač (např. CCD) + výkonný zesilovač



Jednoduché systémy využívající CCD prvky snímají záření kolem 1mm

Všechny nevýhody CCD snímání

## Zpracování obrazu a signálu – Histogram obrazu

### Obsah

Zpracování obrazu a signálu – Histogram obrazu .....	1
Histogram obrazové matice .....	2
Výpočet histogramu .....	3
Úkol – výpočet histogramu.....	3
Úkol - vizualizace histogramu obrazové matice (01 histogr_vizual) .....	3
Ekvalizace histogramu .....	4
Úkol ekvalizace histogramu (02histogramEqu).....	4
Domácí úkol (03histogramEqu_RGB) .....	5

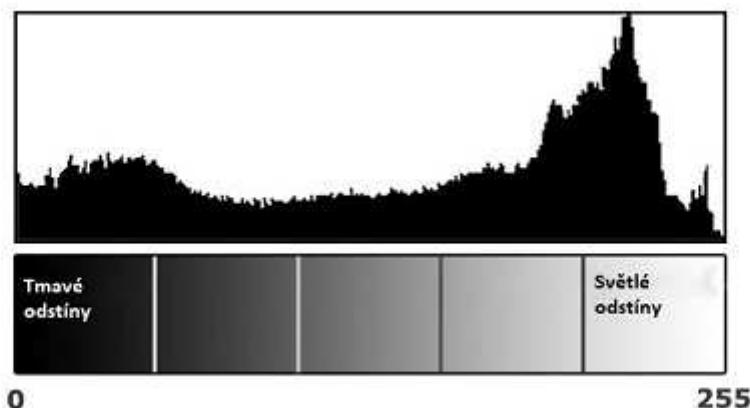
## Histogram obrazové matice

- Histogram je statistické vyjádření distribuce dat v náhodném výběru.

Jedná se o diskrétní funkci  $h(f_k) = n_k$ ,

Kde  $f_k$  je k-tá úroveň jasu a  $n_k$  je počet pixelů v obrazu s úrovní  $f_k$ .

- Obvyklé je vyjádření diskrétně pomocí sloupcového grafu se sloupci stejné šířky, vyjadřující šířku intervalů (tříd).
- Výška sloupců vyjadřuje četnost (či pravděpodobnost výskytu) sledované veličiny v daném intervalu.



Ve zpracování obrazu je histogram používán pro grafické znázornění distribuce jasu ve snímku.



Histogram je možné použít např. i pro prahování (viz. další kapitoly).

## Výpočet histogramu

Výpočet histogramu je velmi prostý – obvykle jej realizujeme jako jednorozměrné pole s položkami  $h[0] .. h[L-1]$ .

Následně je procházena celá matice obrazu a pro každý obrazový bod  $p(x,y)$  je inkrementována příslušná hladina pole  $h[]$ .

Na konci algoritmu pole  $h[]$  reprezentuje četnosti jasu. Pokud histogram vydělíme celkovým počtem bodů v obraze  $p(f_k) = n_k/N$ , přejde histogram na funkci pravděpodobnosti výskytu jednotlivých úrovní jasu v obrazu.

## Úkol – výpočet histogramu

- Vytvořte funkci (třídu) `private void Statistic();` která zabezpečí výpočet histogramu prvního barevného (jasového) kanálu.
- Histogram ukládejte do globálně definovaného pole `int[] Histogram = new int[256];`
- V cyklu projděte všechny body obrazu, vyčtěte jas a každému jasu (jasové hodnotě) v poli `HistoGram_cely[c.R]` zvýšte hodnotu o 1:

```
Color c = bm.GetPixel(x, y);
Histogram[c.R] = Histogram[c.R] + 1;
```

- Spusťte výpočet, umístěte *breakpoint* za tělo výpočtu a sledujte vypočtené hodnoty:
  - Na pole `Histogram[]` „přidejte kukátko add Watch“.
  - V kukátku zvolte vybrat vše, kopírovat a vložte do Excelu.
  - Sečtěte celkový počet hodnot v poli. Čemu odpovídá?
  - **Zobrazte ve formě grafu.**

## Úkol - vizualizace histogramu obrazové matice (01 histogr\_vizual)

- Do aplikace přidejte `pictureBox` a nastavte jeho šířku na 256px.
- Dále přidejte funkci (třídu) `DrawHistoGram()`, která z vypočtených dat vykreslí do `pictureBox` histogram pro první kanál snímku.

Třída `DrawHistoGram`:

```
private void DrawHistoGram() //vstupem pole vzdy 256 dlouhe!!!
{
    int pocet = 256;    //pocet sloupecku (trid) histogramu
    Bitmap histogram_temp = new Bitmap(pictureBox_Histogram.Width, pictureBox_Histogram.Height);
    Graphics g = Graphics.FromImage(histogram_temp);
    g.Clear(Color.White);

    //smazu pripadny stary histogram
    pictureBox_Histogram.Image = null;
    pictureBox_Histogram.Update(); //tohle je dulezite. Bez toho se neprkresli

    //vytvorim pera. Jedno pro osy, jedno plne pro data
    Pen Pen_axes = new Pen(Color.Red);
    Pen Pen_data = new Pen(Color.Lime); //Red, Peru, Lime, Khaki...

    int tloustka_sloupce = pictureBox_Histogram.Width / pocet;
    //mam min 256 px sirokej histogram, tak nejmensi cara nebude nikdy uzsi nez 1!

    int h = pictureBox_Histogram.Height;
    //vyhledani maxima
    int hist_cely_max = 0;
```

```

for (int i = 0; i < 256; i++) if (Histogram[i] > hist_cely_max) hist_cely_max = Histogram[i];

//normalizace histogramu na vysku grafu
int[] HistoGram_cely_normalized = new int[256];
for (int i = 0; i < (256); i++)
{
    HistoGram_cely_normalized[i] = (Int32)((double)Histogram[i] / (double)(hist_cely_max / (double)h));
}

//vykreslim histogram celeho obrazku
for (int i = 0; i < (pocet); i++)
{
    double hodnota = HistoGram_cely_normalized[i*(256/pocet)];
    g.DrawRectangle(Pen_data, i * tloustka_sloupce, h - System.Convert.ToInt32(hodnota), tloustka_sloupce, h);
}

//vykreslim osy
g.DrawRectangle(Pen_axes, 1, 1, 1, h); //vertikalni
g.DrawRectangle(Pen_axes, 1, h - 1, pictureBox_Histogram.Width, h - 1); //horizontalni
hist_ymax_label.Text = hist_cely_max.ToString();

pictureBox_Histogram.Image = histogram_temp;
}

```

Prozkoumejte histogram a interpretujte jeho význam.

Změňte hodnotu proměnné `int pocet`, např. na 64 a pozorujte výsledek.

## Ekvalizace histogramu

Pokud obraz nemá jasové úrovny rovnoměrně rozděleny v celém rozsahu, či nevyužívá celkový rozsah, jeví se uživateli jako nekontrastní, pod, či přeexponovaný.

- Existuje více způsobů ekvalizace histogramu – v závislosti na předpokládaném rozdělení pravděpodobnosti transformační funkce (viz. kurz Zpracování obrazu).
- Nejčastějším předpokladem je požadavek na rovnoměrné rozšíření v rozsahu  $g_0 - g_k$ , kde  $g_0$  představuje počáteční a  $g_k$  koncovou hodnotu **nové stupnice** jasu transformovaného obrazu. Pak je ekvalizace provedena podle vztahu:

$$g = T(f) = \frac{g_k - g_0}{N^2} \sum_{i=f_0}^f H_f(i) + g_0$$

Pozor!  $-\sum_{i=f_0}^f H_f(i)$  je představována součtem intenzit v histogramu (nejčastěji od 0 do  $i$ -té intenzity)

## Úkol ekvalizace histogramu (02histogramEqu)

- Vytvořte funkci (třídu) `HistEq()`, která ekvalizuje obrazovou matici za pomocí předchozího vypočteného histogramu.
  1. Procházejte celý obraz po pixelech
  2. V každém bodě vypočtěte samostatně sumu  $H...$
  3. Vypočtěte novou hodnotu jasu  $g$
  4. Pracujte nejprve v prvním kanálu, poté přidejte další.

Pozor! – pracujete-li v `Int`, proveděte nejprve násobení  $(g_k - g_0) * suma H$  a až poté dělte  $N$ , jinak by výsledkem dělení bylo vždy číslo menší, než 1 a tedy v `Int` číslo 0!

- Nově transformovaný obraz vykreslete.
- A spočítejte jeho nový histogram.

#### Domácí úkol (03histogramEqu\_RGB)

- Třídu `Statistic()`; rozšiřte tak, že výpočet provedete i pro 2. a 3. kanál.
- Třídu `DrawHistoGram()`; rozšiřte tak, že vykreslí histogramy všech 3. kanálů – barevně odlišené.
- Třídu `HistEq()`; rozšiřte pro práci ve 3 kanálech (pro každý kanál se využije vlastní histogram)

## Konvoluční filtrace

### Obsah

Konvoluční filtrace.....	1
Konvoluce .....	2
Výpočet konvoluce .....	2
2D diskrétní konvoluce.....	3
Úkol (1) –2D konvoluce v Matlabu .....	4
Úkol (2) –Vlastní implementace 2D konvoluce v Matlabu .....	4
Implementace v C#.....	5
Aplikace konvoluce (DU) .....	5

## Konvoluce

Konvoluce je matematický operátor zpracovávající dvě funkce.

Spojitá konvoluce jednorozměrných funkcí  $f(x)$  a  $h(x)$  je definována vztahem:

$$W(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\alpha)h(x - \alpha)d\alpha,$$

V diskrétní podobě pak:

$$W(x) = f(x) * h(x) = \sum_{\alpha} f(\alpha)h(x - \alpha)d\alpha$$

Funkce  $g(x)$  je nazývána jádro (kernel) např. lineární transformace.

V našem případě bude např.  **$f(x)$**  představovat např. obrázek a  **$h(x)$**  např. imp. odezvu filtru.

**Konvoluce je velmi důležitý operátor**, protože (tzv. konvoluční teorém):

$$F(f(x) * h(x)) = F(f(x)) \cdot F(h(x)) = F \cdot H$$

Jinými slovy:

*Signál (obraz) v konvoluci s impulzní odevzvou filtru = součin spektra signálu (obrazu) s filtrem!*

$g(x) / G(x)$  mohou představovat např.:

- dolnofrekvenční filtr k vyhlazení signálu
- jádra lineárních transformací (např. FT)
- hranový operátor k nalezení hran v obrazu
- efekty aplikované na obrázek
- transformace obrazové matice
- jádro DC transformace pro kompresi v JPEG
- a mnoho dalších činností v oblasti zpracování signálu/obrazu...

## Výpočet konvoluce

Výpočet (v tzv. prostorové/časové) oblasti provádíme dle vztahu:

$$W(x) = \sum_{\alpha} f(\alpha)h(x - \alpha)d\alpha$$

Příklad:

$$W(0) = f(0) \cdot g(0)$$

$$W(1) = f(0) \cdot g(1) + f(1) \cdot g(0)$$

$$W(2) = f(0) \cdot g(2) + f(1) \cdot g(1) + f(2) \cdot g(0)$$

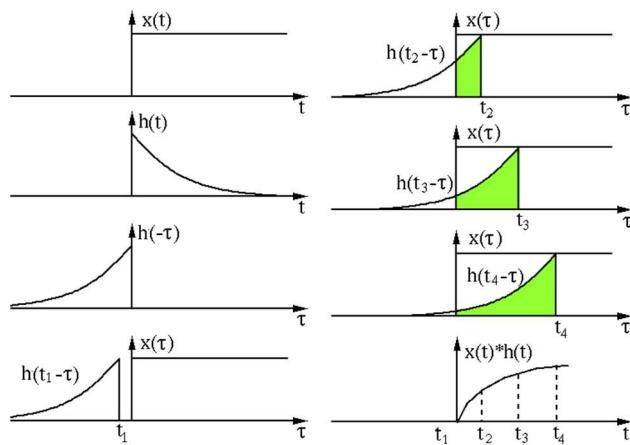
...

$$W(n) = f(0) \cdot g(n) + f(1) \cdot g(n-1) + \dots + f(n) \cdot g(0)$$

...

$$W(2 \cdot n - 1) = f(n) \cdot g(n)$$

Důležitý poznatek – funkce  $h(x)$  se nasouvá „jakoby“ odzadu, resp. obráceně.



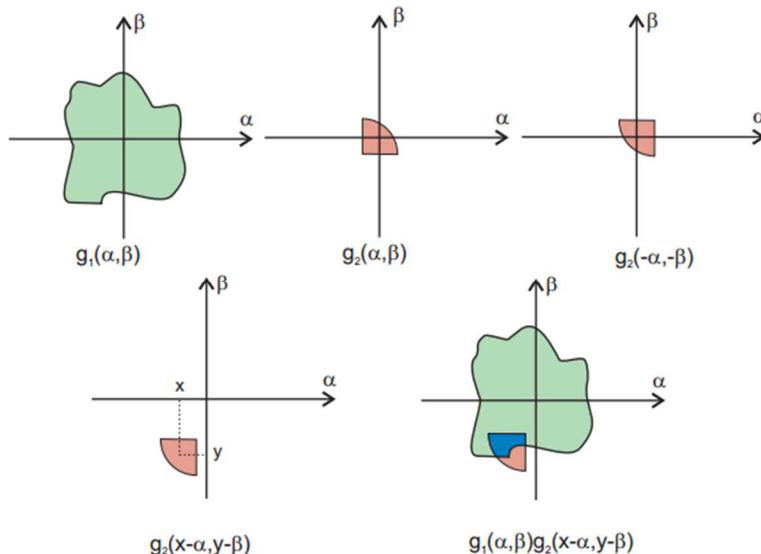
## 2D diskrétní konvoluce

Výpočet konvoluce může být rozšířen pro vícerozměrné signály:

$$f(x, y) * h(x, y) = \sum_{\alpha} \sum_{\beta} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta$$

Zpracování je shodné s výpočtem 1D konvoluce.

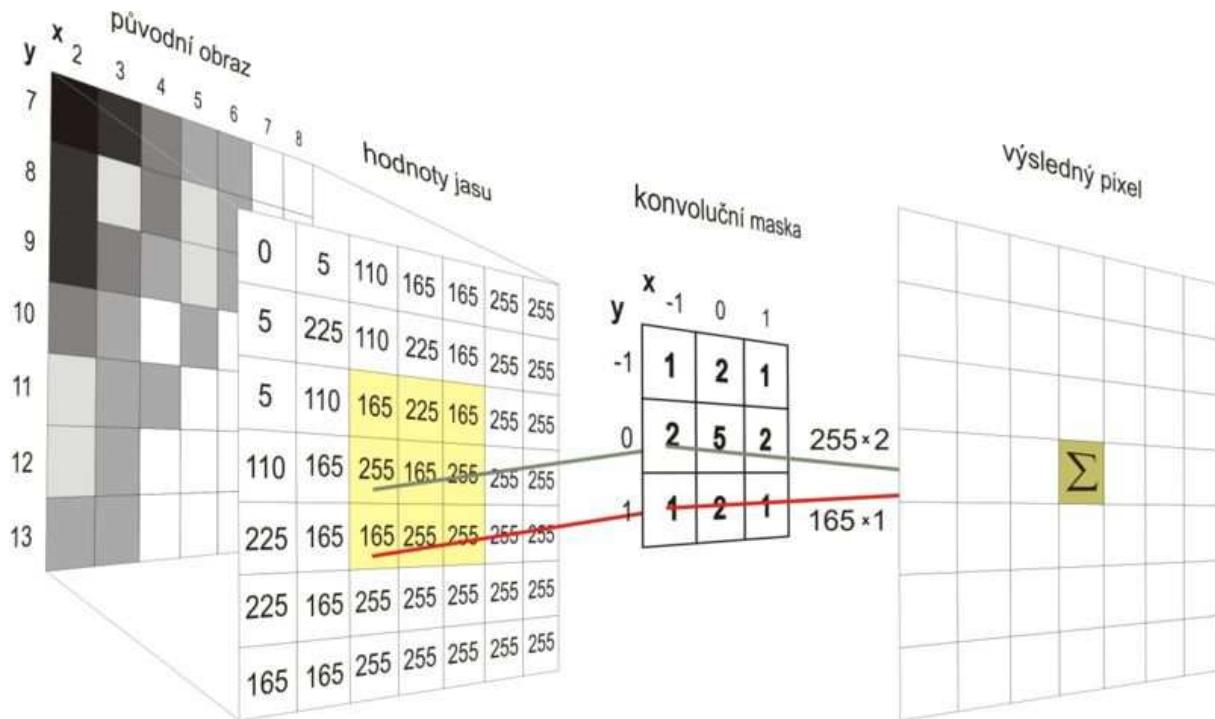
Pozor! – funkce  $h(x, y)$  je opět otočena a to v obou souřadnicích!



Pro většinu jader nepodstatné (jádra např. filtrů jsou obvykle definována symetricky...).

V některých případech se však jedná o důležitý poznatek (např. při metodě Template-Matching).

- Výpočet konvoluce je lokální operace nad obrazovým bodem (pixelem) a jeho bezprostředním okolím!
- Hodnota centrálního pixelu je vlastně *vážená suma okolí*.
  - Váhy jsou dány otočeným 2D konvolučním jádrem.



### Úkol (1) – 2D konvoluce v Matlabu

1. Načtěte v Matlabu snímek *moon.tif* (součást Matlabu) a převeďte do formátu double <0;1>
2. Vytvořte následující kernel pro vyhlazení obrazové matice:

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

3. Prostudujte v návodě použití příkazu *conv2* v Matlabu.
  4. Aplikujte pomocí *conv2* konvoluční jádro h na obrázek a zobrazte výsledek.
    - použijte volání funkce *conv2(O,h,'valid')*;
  5. Vytvořte kernel pro zaostření obrazové matice:
- $$h = \frac{1}{6} \begin{bmatrix} -1 & -4 & -1 \\ -4 & 26 & -4 \\ -1 & -4 & -1 \end{bmatrix}$$
6. Aplikujte pomocí *conv2* konvoluční jádro h na obrázek a zobrazte výsledek.
  7. Diskutujte výsledky

### Úkol (2) – Vlastní implementace 2D konvoluce v Matlabu

1. Vytvořte funkci *function [ O2 ] = konv2(O,h)*, ve které implementujete výpočet 2D diskrétní konvoluce.
2. Proveďte s funkcí *konv2* tytéž kroky, jako v předchozím úkolu a porovnejte výsledek.

Pro ověření správnosti výpočtu můžete použít následující příklad:

$$O = \begin{bmatrix} 1 & 8 & 15 \\ 7 & 14 & 16 \\ 13 & 20 & 22 \end{bmatrix} \quad h = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \quad V = 575$$

### Implementace v C#

1. Implementujte do programu z minulých bloků třídu konv2, která provede výpočet 2D konvoluce.

Pozor! na rozdíl od MATLABu C# používá jinak orientovaný souřadný systém.
2. Přidejte tlačítko, které použije třídu konv2 separátně na všechny 3 kanály.
3. Kernel definujte v číselném formátu *float* a omezte rozsah hodnot transformované matici.

```
if (R2 < 0) R2 = 0; if (R2 > 255) R2 = 255;
```

### Aplikace konvoluce (DU)

1. Vymyslete a implementujte způsob pro pohodlné zadávání kernelů dalších efektů.

Myšlenka:

Použijte např. komponentu *dataGridView*

a k datům přistupujte např.: *kernel[i] = System.Convert.ToDouble(kernel\_dataGridView.Rows[i].Cells["coefficient"].Value);*  
nebo něco podobného.

Uvažujte rozměr kernelu max. 5 x 5 koeficientů.

2. Vyzkoušejte jádra transformací z: [https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))
3. Vzhledem k délce výpočtu přidejte inf. pro uživatele do stavového řádku o progresi algoritmu.

## Konvoluční filtrace - pokračování

### Obsah

Konvoluční filtrace - pokračování .....	1
Konvoluce .....	2
Filtrace obrazu (cv01.m) .....	2
Vyhlažování průměrováním .....	2
Binomický průměr .....	2
Gaussovo vyhlazování (cv03.m) .....	3
Filtrace obrazu – horofrekvenční průběh (cv02.m).....	3
Hranová detekce – segmentace prahováním (cv04.m).....	4
Hranová detekce (cv05.m) .....	4
DU .....	5
Další téma.....	5

## Konvoluce

Z minulého bloku: **konvoluce je velmi důležitý operátor**, platí:

$$F(f(x) * h(x)) = F(f(x)) \cdot F(h(x)) = F \cdot H$$

Jinými slovy:

*Signál (obraz) v konvoluci s impulzní odezvou filtru = součin spektra signálu (obrazu) s filtrem!*

$g(x) / G(x)$  mohou představovat mnoho zajímavých nástrojů pro úpravy signálu.

## Filtrace obrazu (cv01.m)

### Vyhlažování průměrováním

Vytvořte impulzní odezvu ( jádro) počítající aritmetický průměr z obrazové matice. Jádro bude mít velikost  $3 \times 3$  px.

Aplikujte jádro na obrazovou matici a porovnejte výsledek před a po operaci.

Odpovězte na následující otázky:

- K jakému efektu došlo?
- Jakého typu je aplikovaný filtr.
- **Vypočtěte spektrum filtru a zobrazte**
  - Použijte větu o změně čas. měřítka.
  - Použijte prostřední řádek  $h$  a 1D fft.

Rozšiřte jádro na  $5 \times 5$  px a porovnejte výsledek.

### Binomický průměr

Aplikujte tzv. binomický filtr podle následujícího předpisu a pozorujte výsledek.

Porovnejte průběh spektra filtru.

#### *Binomický průměr*

$$h_1 = \frac{1}{16} (1,4,6,4,1)$$

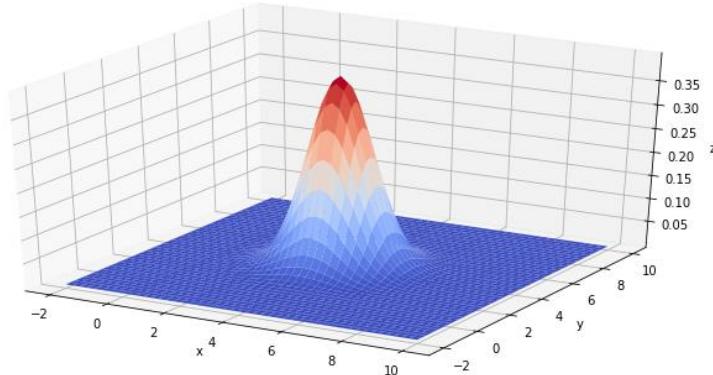
- převod do 2D

$$h = h_1^T * h_1 = \frac{1}{16} \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix} * \frac{1}{16} (1,4,6,4,1) = \frac{1}{256} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

### Gaussovo vyhlazování (cv03.m)

Gaussovo vyhlazování je definováno kernelem approximujícím Gaussovský 2D průběh rozdělení pravděpodobnosti. Koeficienty blíže středu mají vyšší váhu a odpovídají hodnotám na Gaussově křivce definované:

$$h(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Výhodou použití Gaussovského filtru je jemnější přechod mezi oříznutou částí a propuštěnou částí spektra signálu.

Úkol:

1. Implementujte pomocí Matlabu výpočet  $h(x)$  ( $\mu = 0$ ,  $\sigma = 0.3$ )
2. Zobrazte ve formě grafu.
3. Approximujte pro velikost vektoru 5 bodů.
4. Pomocí rotace  $(h' \times h) / \text{sum2}(h' \times h)$  převeďte na 2D impulzní odezvu.
  - získané koeficienty zadejte do aplikace v C# a aplikujte na obraz.
5. Aplikujte na obrazovou matici a zobrazte spektrum filtru.
6. Experimentujte se sigma od 0.3 až např. 1.5

### Filtrace obrazu – hornofrekvenční průběh (cv02.m)

Aplikujte následující filtr:

$$h = \begin{bmatrix} -0 & -1 & -0 \\ -1 & 5 & -1 \\ -0 & -1 & -0 \end{bmatrix}$$

Analyticky zkuste odhadnout, jak filtr pracuje (které pixely zdůrazní a co naopak potlačí)

Aplikujte jádro na obrazovou matici a porovnejte výsledek před a po operaci.

Odpovězte na následující otázky:

- K jakému efektu došlo?
- Jakého typu je aplikovaný filtr.
- **Vypočtěte spektrum filtru a zobrazte**

- Použijte větu o změně čas. měřítka.
- Použijte prostřední řádek  $h$  a 1D  $fft$ .

Zvyšte strmost filtru a opakujte předchozí postup:

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

## Hranová detekce – segmentace prahováním (cv04.m)

Detekce hran je postup v digitálním zpracování obrazu, sloužící k nalezení oblastí pixelů, ve kterých se podstatně mění jas.

- Předpokládáme obvykle, že nejrychlejším změnám dochází na hranicích předmětů/objektu v obrazu, či výskytu užitečného signálu např. v audiozáznamu.

Hranová detekce sestává ze dvou kroků:

1. Zvýraznění nespojitostí jasové funkce →
2. prahování

Elementární, avšak nepříliš účinná metoda je přímá segmentace na základě stanovení prahu.

1. Načtěte v Matlabu obrázek *picture-002.jpg* a převeďte jej do formátu double a odstínů šedi.
2. Pomocí funkce *imhist* zobrazte histogram a diskutujte jeho význam.
3. Pokuste se zvolit hodnotu prahu, kterou na základě znalosti histogramu oddělíte objekty od pozadí.
4. Implementujte do C# aplikace metodu, která na základě ručně stanoveného prahu provede binární segmentaci obrazu – objekty budou vyjádřeny hodnotou 255 a pozadí hodnotou 0.
5. Opakujte totéž na obrázku „patvar.jpg“  
Pokuste se nalézt hodnotu prahu, která by spolehlivě vyseparovala celý předmět od pozadí.
6. Analyzujte, proč se separace prahováním v tomto případě nedáří.

## Hranová detekce (cv05.m)

1. Pokuste se na obrázek aplikovat nejprve následující jádro:

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

2. Vypočtěte spektrum prvního řádku matice h a vysvětlete jeho význam.
3. Prohlédněte si výsledek a využijte funkci *improfile* v Matlabu k analýze.
4. Nyní aplikujte nově nalezenou hodnotu prahu a obraz zbinarizujte.
5. Orotujte matici h o 90° a znova aolikujte.

## DU

6. Aplikujte celý postup (s ručním zadáním prahu) do aplikace v C# jako metodu prahování.
7. Vyzkoušejte další hranové operátory:

-1	-1	-1
0	0	0
1	1	1
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-1	0	1
-1	0	1
-1	0	1
-2	0	2
-1	0	1

→ *operátory Prewittové*<sup>4</sup>

→ *Sobelovy operátory*

(za cenu menší citlivosti - provádí i filtraci)

1	1	1
1	-2	1
-1	-1	-1

1	1	1
-1	-2	1
-1	-1	1

-1	1	1
-1	-2	1
-1	1	1

→ *operátory Robinson*

3	3	3
3	0	3
-5	-5	-5

3	3	3
-5	0	3
-5	-5	3

5	3	3
-5	0	3
-5	3	3

→ *operátory Kirsch*

## Další téma

- Otsu's metoda.
- Prahování s distribucí chyby.
- Mediánová filtrace.