Multimédia 2 (MM 2)

Termín: Letný semester 2018/2019
Rozsah: 2h prednášky, 2h cvičenia

Prednášateľ & vedúci cvičení: Ing. Radoslav Vargic, PhD. (B 415)

Kontakt (prednášajúci):

Miestnosť: B415
Tel: 02 60291 788

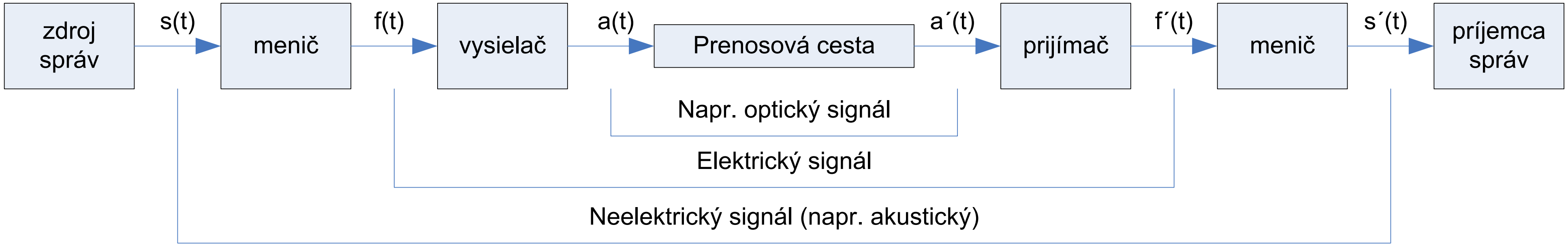
E-mail: radoslav.vargic@stuba.sk
WWW: http://www.ktl.elf.stuba.sk/study/mm2/

Ostatné kontakty (ICQ, Skype, ...) sú na WWW: http://www.ut.fei.stuba.sk/~vargic

Plán prednášok a cvičení
(viď PDF)

Multimediálny signál / multimediálna sústava

Príklad sústavy



V MM2:

- Nevyhneme sa analógovej forme signálu
- Dôraz pri spracovaní bude na digitálnu formu signálov
- Začneme zopakovaním transformácie a skalárneho súčinu
- Následne prejdeme AD/DA prevod
- Následe prevzorkovanie digitálneho signálu
- Následne systematicky audio, obraz, video ...

Transformácia signálu

- Transformácia mení signál (transformuje ho) aby bol následne vo vhodnejšom tvare/forme vzhľadom na to, čo chceme vykonať
- Rôzne prípady použitia (use cases):

a) **Podobnosť s bázovými funkciami**: Chcem zistiť ako sa signál podobá na mnou stanovené funkcie a potom ho budem posudzovať z hľadiska uvedenej podobnosti
   i) videnie z pohľadu skalárneho súčinu, bázových funkcií - frekvenčná analýza, spracovanie audia, obrazu, videa

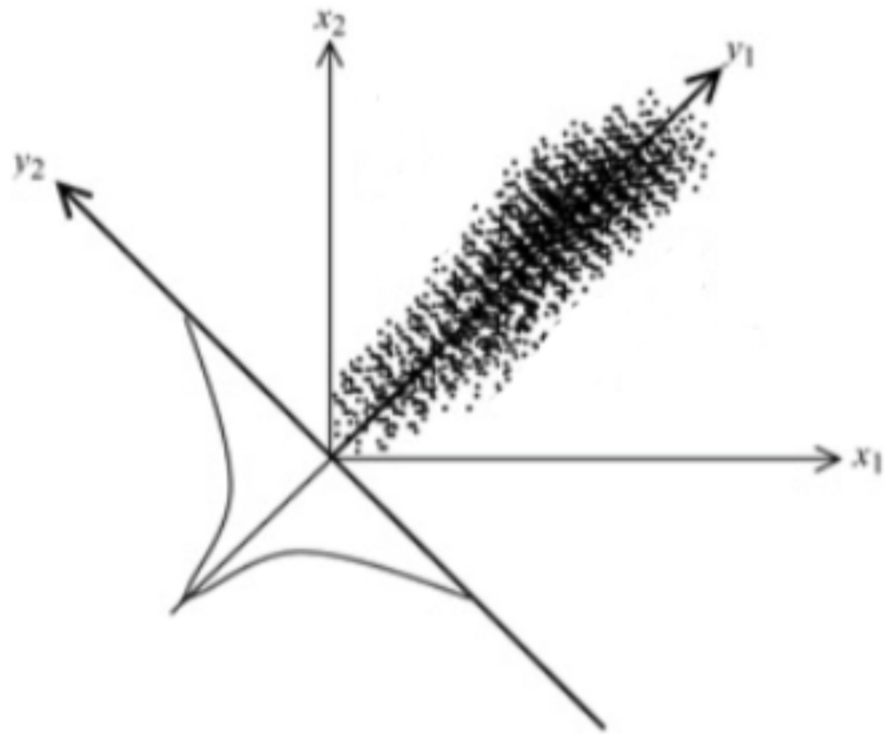
b) **Dekorelovanie**: Signál som „dekoreloval“ – odstránil som v ňom určité závislosti
   i) Napr. ak sú súradnice dvojice bodov rozmiestnené nejako v priestore tak sa na ten priestor viem lepšie pozrieť
   ii) Videnie z pohľadu štatistiky, spracovanie audia, obrazu, videa

c) **Súradnicový systém**: Získal som súradnice signálu v inom súradnicovom systéme (toto videnie je dôležité pri počítačovej grafike)

Prípad „a“: Podobnosť s bázovými funkciami

- Podobá sa náš signál na vlnu? Vlnový balík? Šum? Kocky?
- Transformácia pomocou Fourierovej transformácie
    - základné druhy fourierovej transformácie – **opakovanie (viď PDF)**
- Ešte samozrejme aj kosínusové, sínusové, waveletové, Walshove, ... transformácie

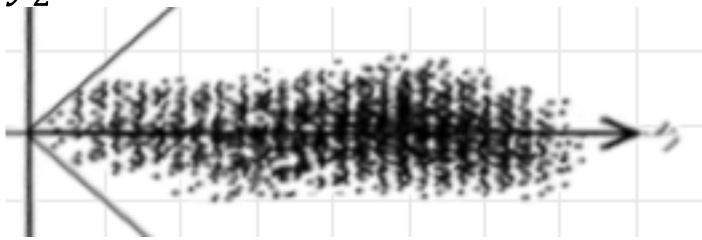
Prípad „b“: Odstránenie závislostí (dekorelácia)



Umiestnenie dvojíc bodov má závislosť.
Uvedené body môžu predstavovať napríklad:

- Súradnice udalostí, ktoré si je treba zapamätať.
- Susedné hodnoty v nejakom signále (audio, obraz, ...)

Transformácia pomôže presnejšiemu zapamätaniu si. Ako? Súradnicový system otočíme a použijeme y_1 a y_2:



Vidíme, že údaje zaberajú 8x2 štvorčekov. T.j. s nejakou presnosťou vieme lokalizovať každý bod a potrebovali sme na to 4 (3+1bit). Keby sme nechali priestor tak ako bol pôvodne, potrebovali by sme na to 6 bitov (3+3bity), pričom väčšina kombinácií by ostala nevyužitá.

Odstránenie závislostí 2

- Podobne ako sme to urobili v 2D prípade sa dá ísť do vyšších rozmerov a odstraňovať závislosti v N-ticiach koeficientov (N-tica -> veľkosť bloku) pri blokovej transformácii.
- Dá sa nájsť "optimálna transformácia pre daný signal" (pomocou korelácie/kovariancie)
- Takáto transformacia sa volá Karhunen-Loevova transformácia (KLT)
- Pri kompresii to ale nepomôže, lebo si je treba zapamätať aj celú transformačnú maticu ...
- Podobné vlastnosti má ale aj napr. DCT (Diskrétna kosínusová Transformácia), ktorá je fixná, takže sa v praxi používa najmä tá (kompresia audia, kompresia videa)
- Pri kompresii obrazu sa ale presadil úplne iný typ transformácie – Waveletová transformácia (dozviete sa na predmete PMSOV)

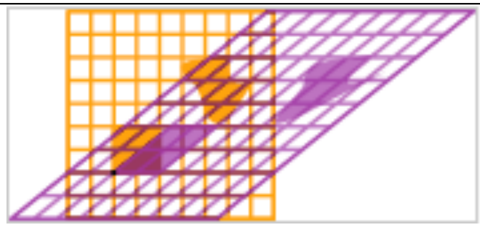
Prípad „c“: Súradnicový system

- Ako konkrétne sa otáčali body pri dekorelovaní?

$\begin{pmatrix} x_{new} \\ y_{new} \end{pmatrix} = \begin{pmatrix} cos\ \phi & -sin\ \phi \\ sin\ \phi & cos\ \phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$	Bod sa otočí okolo počiatku o uhol ϕ (v našom príklade to bolo -45°)

- Hovoríme o geometrických transformáciách
- Interpretujeme buď, že otáčame signály, alebo že otáčame súradnicový systém,
- Resp. transformujeme signály, alebo meníme súradnicový systém

Geometrické transformácie v 2D

$\begin{pmatrix} x_{new} \\ y_{new} \end{pmatrix} = \mathbf{T} \begin{pmatrix} x \\ y \end{pmatrix}$	
rotácia o ľubovoľný uhol	$\mathbf{T} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$
zrkadlenie cez os x x→x, y→-y	$\mathbf{T} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
zrkadlenie cez os y x→-x, y→y	$\mathbf{T} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
Zväčšenie m-krát vo všetkých smeroch (m<1 je zmenšenie)	$\mathbf{T} = \begin{pmatrix} m & 0 \\ 0 & m \end{pmatrix}$
Priemet na os x (x→x, y→0)	$\mathbf{T} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
Priemet na os y (x→0, y→1)	$\mathbf{T} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$
Zmena sklonu (shear) v smere x (m=1 je 45° doprava)	$\mathbf{T} = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix}$ 
Zmena sklonu (shear) v smere y	$\mathbf{T} = \begin{pmatrix} 1 & 0 \\ m & 1 \end{pmatrix}$

- Podobne pomocou matíc 3x3 sa dá transformovať body v 3D priestore

Geometrické transformácie v 2D pomocou 3D (ukážka)

Motivácia: ako pomocou transformácie spraviť posun (transláciu)? Vieme to?

$$\begin{pmatrix} x_{new} \\ y_{new} \end{pmatrix} = \mathbf{T} \begin{pmatrix} x \\ y \end{pmatrix}$$

V 2D to vieme iba takto (pomocou súčtu, čo nie je transformácia):

$$\begin{pmatrix} x_{new} \\ y_{new} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} shift_x \\ shift_y \end{pmatrix} = \begin{pmatrix} x + shift_x \\ y + shift_y \end{pmatrix}$$

Prepísaním do 3D to ale vieme aj pomocou transformácie:

$$\begin{pmatrix} x_{new} \\ y_{new} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & shift_x \\ 0 & 1 & shift_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + shift_x \\ y + shift_y \\ 1 \end{pmatrix}$$

Princíp blokovej transformácie

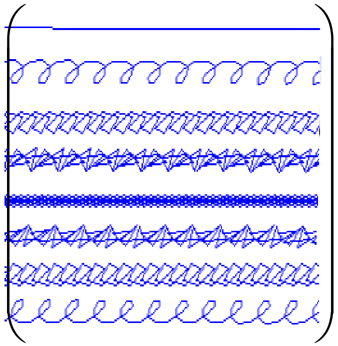
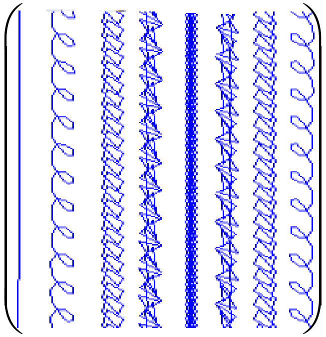
- Zopakujme si princíp z pohľadu lineárnej algebry:
  - pre signál platí, že $x(n) \in C^N$, resp. $x(n) \in R^N$, t.j. jedná sa o priestor všetkých N rozmerných vektorov nad poľom C, alebo R.
  - typická bloková transformácie je napr. DFT (predpokladá periodifikovaný signál $x(n) \in C^N$):

Druh	Pôvodný zápis	Maticový zápis
DFT	$X(k) = \sum_n x(n) W_N^{-kn}$ $W_N = e^{j\frac{2\pi}{N}}$	$\mathbf{X} = \begin{pmatrix} X(0) \\ \dots \\ X(N-1) \end{pmatrix} = \begin{pmatrix} W_N^{-00} & \dots & W_N^{-0(N-1)} \\ \dots & \dots & \dots \\ W_N^{-(N-1)0} & \dots & W_N^{-(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} x(0) \\ \dots \\ x(N-1) \end{pmatrix} = \mathbf{DFT}_N \mathbf{x}$
DFT^{-1}	$x(n) = \frac{1}{N} \sum_k X(k) W_N^{kn}$ $W_N = e^{j\frac{2\pi}{N}}$	$\mathbf{x} = \frac{1}{N} \begin{pmatrix} W_N^{00} & \dots & W_N^{0(N-1)} \\ \dots & \dots & \dots \\ W_N^{(N-1)0} & \dots & W_N^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} X(0) \\ \dots \\ X(N-1) \end{pmatrix} = \mathbf{DFT}^{-1}{}_N \mathbf{X}$

Spomeňme ešte operáciu skalárneho súčinu:

$$\langle f(n), g(n) \rangle = \sum_n f(n) \bar{g}(n)$$

Grafická interpretácia násobenia matice vektorom

Druh	Maticový zápis - graficky	Interpretácia maticového zápisu
DFT	$\begin{pmatrix} X(0) \\ \cdots \\ X(N-1) \end{pmatrix} =$  $\begin{pmatrix} x(0) \\ \cdots \\ x(N-1) \end{pmatrix}$	$\mathbf{X} = \mathbf{DFT}_N \mathbf{x}$ $X(k) = \left\langle DFT_{N,riadok\,k}(n), x(n) \right\rangle$ ako sa „podobá“ signál na jednotlivé riadky matice (**konjugované bázové funkcie**)?
DFT^{-1}	$\begin{pmatrix} x(0) \\ \cdots \\ x(N-1) \end{pmatrix} =$  $\begin{pmatrix} X(0) \\ \cdots \\ X(N-1) \end{pmatrix}$	$\mathbf{x} = \mathbf{DFT^{-1}}_N \mathbf{X}$ $x(n) = \sum_k X(k) DFT^{-1}_{N,stĺpec\,k}(n)$ ako sa dá signál poskladať z váhovaných stĺpcov matice (**bázových funkcií**)?

Pozn.: bázové funkcie sú

Pojem **bázy**: množina vektorov v stĺpcoch matice DFT^{-1} tvorí bázu priestoru C^Nresp. R^N, lebo:

- táto množina vektorov (**bázové funkcie**) je **lineárne nezávislá**
- ich **lineárnou kombináciou** vieme vyskladať ľubovoľný že $x(n) \in C^N$, resp. $x(n) \in R^N$
- jedná sa dokonca o **ortogonálnu bázu**, lebo **vektory sú navzájom kolmé** !

Obraz – 2D signál – matica

- zjedodušme obraz (obrázok) momentálne na jasovú zložku
- hodnota každého obrazového bodu je potom miera jasu
- obrazy sú spravidla obdĺžniky – množiny MxN bodov (dvojrozmerné pole)
- obraz potom vieme reprezentovať ako maticu (a jeho spektrum tiež!)

$$x(n_1,n_2) = \begin{pmatrix} x(1,1) & \dots & x(1,n_2) \\ \dots & \dots & \dots \\ x(n_1,1) & \dots & x(n_1,n_2) \end{pmatrix}, X(k_1,k_2) = \begin{pmatrix} X(1,1) & \dots & x(1,k_2) \\ \dots & \dots & \dots \\ X(k_1,1) & \dots & X(k_1,k_2) \end{pmatrix}$$

Vyvstávajú otázky:

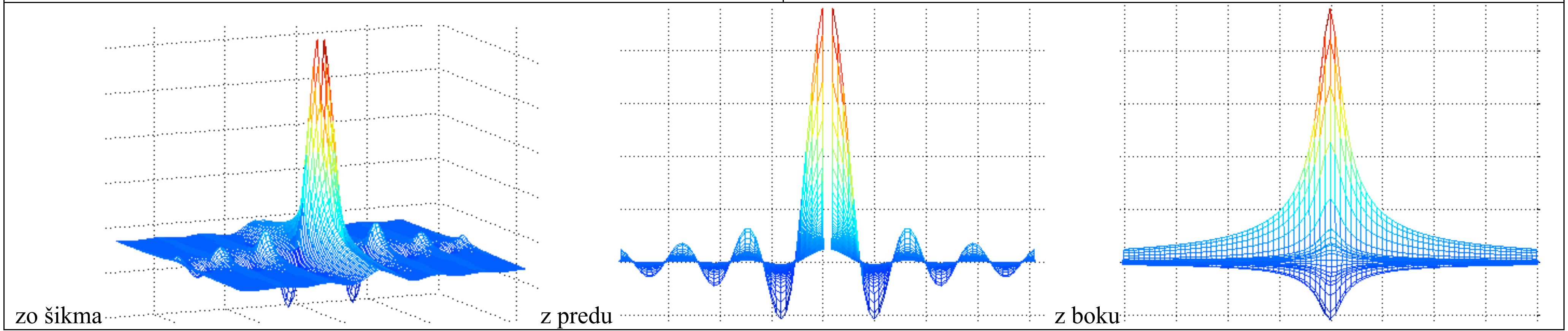
- Ako získame spektrum?
- Ako vyzerajú bázové funkcie?
- Čo je spektrum? To iste čo v prípade 1D signálu
  - množina koeficientov, ktorými sa naváhujú bázové jednotlivé funkcie výsledok sa sčíta a vznikne pôvodný obraz
  - množina koeficientov, ktoré vyjadrujú podobnosť signálu s jednotlivými bázovými funkciami
    - čím väčšia podobnosť, tým väčší koeficient
    - čím menšia podobnosť, tým menší koeficient
    - žiadna podobnosť = sú ortogonálne = ich skalárny súčin je 0
      - $\langle f(n_1,\mathrm{n}_2), g(n_1,n_2)\rangle = \sum_{\mathrm{n}_1} \sum_{n_2} f(n_1,n_2)\bar{g}(n_1,n_2)$

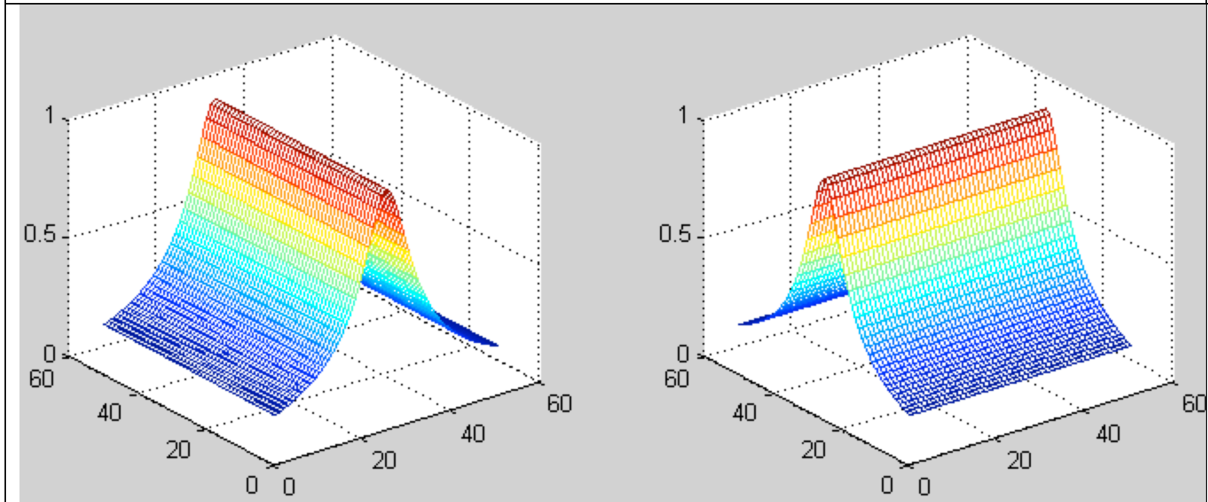
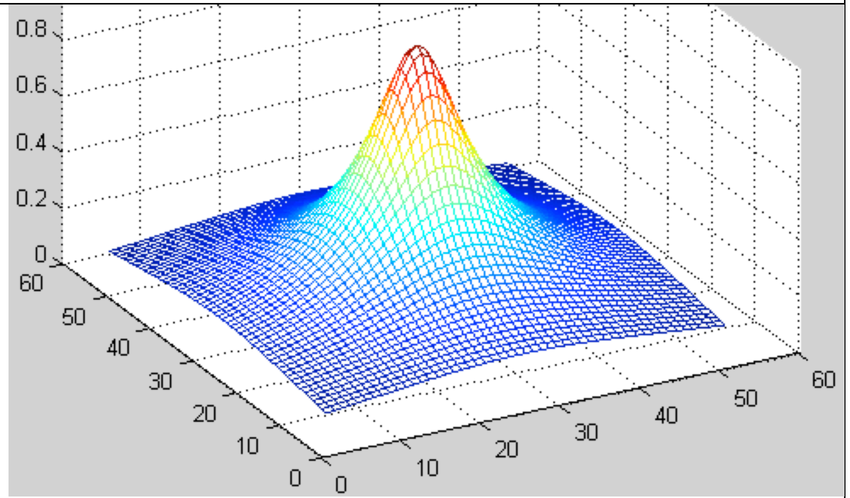
2D transformácia – separovateľný prípad

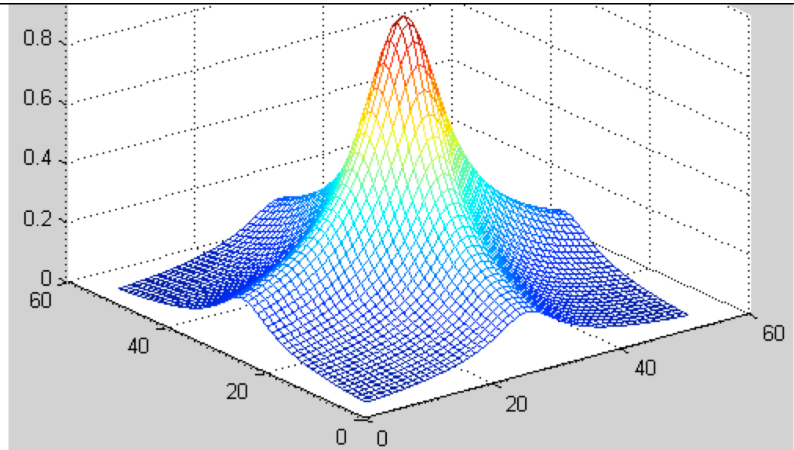
- separovateľný prípad = bázové funkcie sú **separovateľné** funkcie $f_{k_1,k_2}(n_1,n_2) = f_{k_1}(n_1)\, f_{k_2}(n_2)$
- pre lepšiu predstavu, čo je to „separovateľný“ zamyslite sa nad spojitým prípadom a uvážte, či napr. pologuľa nad rovinou xy sa dá vyjadriť ako f(x,y)=g(x)h(y).
    - príklad separovateľnej funkcie: f(x,y)=(sin(x)/x)*(1/sqrt(1+y^2)), táto funkcia vyzerá z jednej strany ako si funkcia a z druhej strany ako pichliač, alebo f(x,y)=(1/sqrt(1+x^2))*(1/sqrt(1+y^2))
    - príklad neseparovateľnej funkcie f(x,y)= 1/sqrt(1+x^2 + y^2), toto sa nedá upraviť do tvaru g(x)h(y).
- Ak sa jedná o tento typ transformácie, potom sa dá transformovať 2D signál najprv „po riadkoch“ a potom „po stĺpcoch (alebo naopak).

f(x,y)=(sin(x)/x)*(1/sqrt(1+y^2)))

```
[X,Y] = meshgrid(-20:.5:20);
Z =sin(X)./X.*1./sqrt(1+Y.^2);
mesh(Z);
```

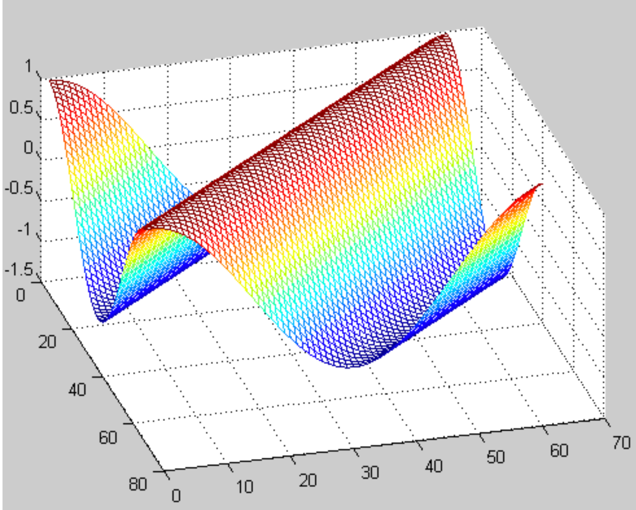


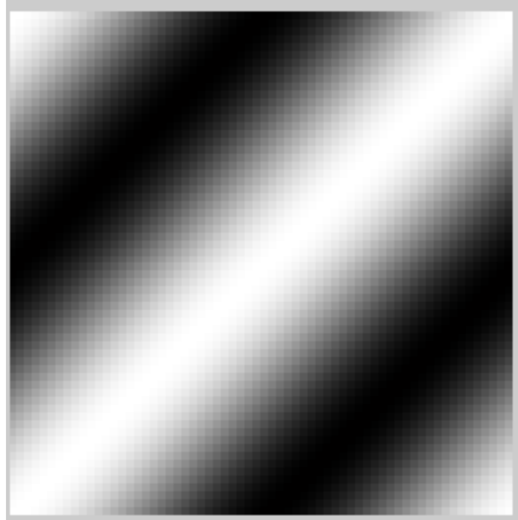
<table>
<tr>
<td>f(x,y)=(1/sqrt(1+x^2))=g(x)
f(x,y)=(1/sqrt(1+y^2))=h(y)</td>
<td>f(x,y)= 1/sqrt(1+x^2 + y^2)</td>
</tr>
<tr>
<td><code>[X,Y] = meshgrid(-5:.2:5);</code>
<code>subplot(1,2,1);Z =1./sqrt(1+X.^2); mesh(Z);</code>
<code>subplot(1,2,2);Z =1./sqrt(1+Y.^2); mesh(Z);</code></td>
<td><code>[X,Y] = meshgrid(-5:.2:5);</code>
<code>Z =1./sqrt(1+X.^2 + Y.^2)</code>
<code>mesh(Z);</code></td>
</tr>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>f(x,y)=g(x)h(y)

<code>[X,Y] = meshgrid(-5:.2:5);</code>
<code>Z =1./sqrt(1+X.^2).*1./sqrt(1+Y.^2);</code>
<code>mesh(Z);</code></td>
<td>
</td>
</tr>
</table>

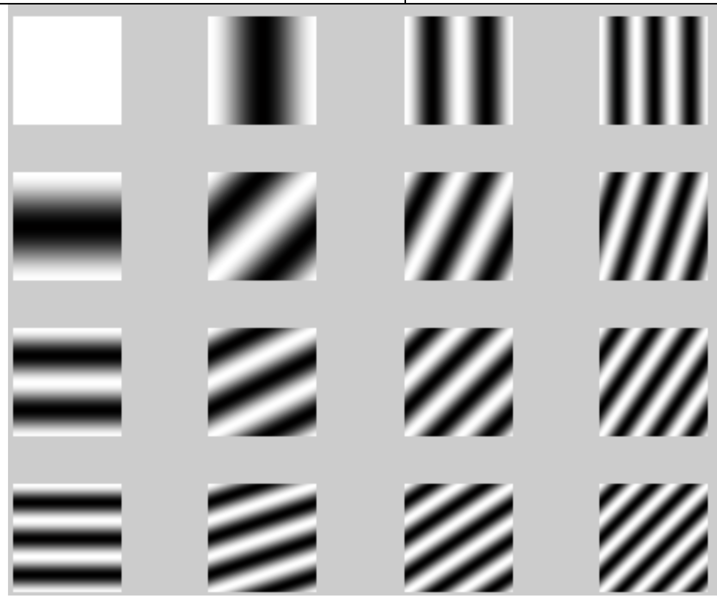
Ako vyzerajú bázové pri 2D DFT? (napr. ich reálna časť)

```
sp=zeros(64);
sp(2,2)=64*64;
cas=ifft2(sp);
mybase=real(cas);
mesh(mybase); figure;
imshow(mat2gray(mybase));
```





```
sp=zeros(64);
for ix=1:4
    for iy=1:4
        sp=zeros(64);
        sp(iy,ix)=64;
        cas=ifft2(sp);
        mybase=real(cas);
        subplot(4,4,4*(iy-1)+ix);
        imshow(mat2gray(mybase));
    end
end
```



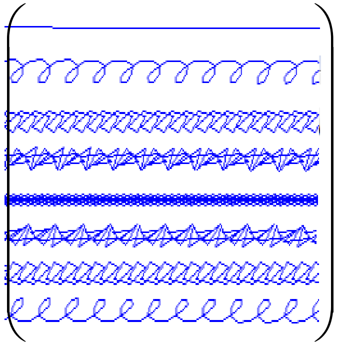
Výpočet 2D DFT

- Budeme počítať spektrálne koeficienty „po jednom“ tak, že budeme zisťovať ako sa podobá na „bázové funkcie“ (plachty) pomocou skalárneho súčinu (funguje aj v neseparovateľnom prípade)
    - $X(k_1, k_2) = \langle f_{k_1,k_2}(n_1, \mathrm{n}_2), x(n_1, n_2)\rangle = \sum_{\mathrm{n}_1} \sum_{n_2} \overline{f_{k_1,k_2}}(n_1, n_2)x(n_1, n_2)$
    - kde f_{k_1,k_2}je príslušná bázová funkcia
    - môže byť na skúške príklad, keď máte zadanú množinu bázových funkcií a pomocou skalárneho súčinu budete musieť získať príslušné spektrálne koeficienty
- „po riadkoch a po stĺpcoch – funguje iba v separovateľnom prípade
    - pomocou maticového počtu (môže byť na skúške pre N=M=2) – viď ďalší slide
    - pomocou rýchlych algoritmov (FFT)
- v Matlabe priamo zavoláte fft2 (ale predtým si prečítate v manuáli, čo a ako to vlastne robí)

Maticový výpočet 2D DFT

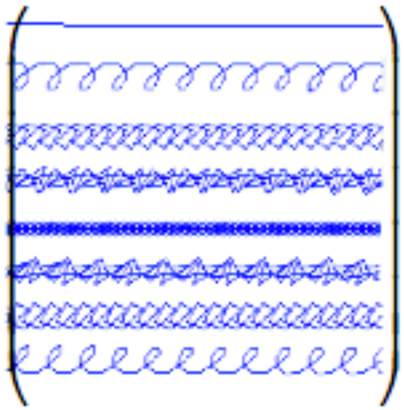
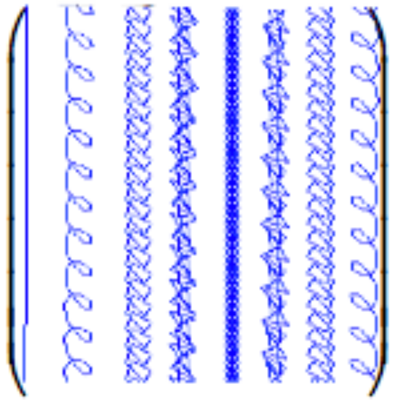
1D DFT: $\mathbf{X} = \mathbf{DFT_N x}$

$$\begin{pmatrix} X(0) \\ \\ \dots \\ \\ X(N-1) \end{pmatrix} =$$



$$\begin{pmatrix} x(0) \\ \\ \dots \\ \\ x(N-1) \end{pmatrix}$$

2D DFT: $\mathbf{X} = \mathbf{DFT_N x \, DFT_N^T}$

<table>
<tr><td>$\begin{pmatrix} X(1,1) & \dots & x(1,k_2) \\ \dots & \dots & \dots \\ X(k_1,1) & \dots & X(k_1,k_2) \end{pmatrix} =$</td><td></td><td>$\begin{pmatrix} x(1,1) & \dots & x(1,n_2) \\ \dots & \dots & \dots \\ x(n_1,1) & \dots & x(n_1,n_2) \end{pmatrix}$</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Transformácia po riadkoch</td></tr>
<tr><td></td><td colspan="2">Transformácia po stĺpcoch</td><td></td></tr>
</table>

Záleží na tom či najprv transformujeme po riadkoch, alebo po stĺpcoch?

- Uvedomte si, že **násobenie matíc je asociatívne**, t.j. platí: ABC=A(BC)=(AB)C

Video – 3D signál

- Interpretujme video ako časovú sekvenciu obrázkou s jasovou zložkou.
- Videá sú potom „kvádre“– množiny MxNxT bodov (trojrozmerné pole
- obraz potom vieme množinu matíc (a jeho spektrum tiež!)

$x(n_1,n_2,n_3) =$	$\begin{pmatrix} x(1{,}1{,}1) & \dots & x(1,n_2,1) \\ \dots & \dots & \dots \\ x(n_1,1{,}1) & \dots & x(n_1,n_2,1) \end{pmatrix}$	$X(k_1,k_2,k_3) =$	$\begin{pmatrix} X(1{,}1{,}1) & \dots & x(1,k_2,1) \\ \dots & \dots & \dots \\ X(k_1,1{,}1) & \dots & X(k_1,k_2,1) \end{pmatrix}$

Vyvstávajú otázky:

- Ako získame spektrum?
- Ako vyzerajú bázové funkcie?
- Čo je spektrum? To iste čo v prípade 1D signálu
  - množina koeficientov, ktorými sa naváhujú bázové jednotlivé funkcie výsledok sa sčíta a vznikne pôvodný obraz
  - množina koeficientov, ktoré vyjadrujú podobnosť signálu s jednotlivými bázovými funkciami
    - čím väčšia podobnosť, tým väčší koeficient
    - čím menšia podobnosť, tým menší koeficient
    - žiadna podobnosť = sú ortogonálne = ich skalárny súčin je 0
      - $\langle f(n_1,\mathrm{n}_2,\mathrm{n}_3), g(n_1,n_2,\mathrm{n}_3)\rangle = \sum_{\mathrm{n}_1}\sum_{n_2}\sum_{n_3} f(n_1,n_2,\mathrm{n}_3)\bar{g}(n_1,n_2,\mathrm{n}_3)$

3D transformácia – separovateľný prípad

- separovateľný prípad = bázové funkcie sú **separovateľné** funkcie
    - $f_{k_1,k_2,k_3}(n_1,n_2,n_3) = f_{k_1}(n_1)f_{k_2}(n_2)f_{k_3}(n_3)$
- Ak sa jedná o tento typ transformácie, potom sa dá transformovať 3D signál najprv „po riadkoch“, „po stĺpcoch“, „po časových slížoch“
- 3D transformácia v maticovom zápise by boli dlhé sekvencie ... nie je dôvod to takto formalizovať
    - stačí robiť FFT2 nad každým obrázkom a následne robiť FFT „po časových slížoch“

- Ako počítať viacrozmernú FFT v Matlabe?
    - „Y = fft(X,[],dim) applies the FFT operation across the dimension dim.“
    - „Y=X.' “ – vykoná transpozíciu matice bez konjugácie
    - 2D FFT: Y = fft(fft(X).').'
    - 2D FFT: Y = fft(fft(X, [], 1), [], 2);
    - 3D FFT: Y = fft(fft(fft(X, [], 1), [], 2), [], 3);
    - nD FFT:

```
Y = X;
for p = 1:length(size(X))
    Y = fft(Y,[],p);
end
```

Transformácia - zhrnutie

- Využitie z hľadiska druhu signálu signálu
  - pre 1D prípad (audio signál, rečový signál,)
  - pred 2D prípad (obraz, 2D plachta v 3D priestore)
  - pre 3D prípad (video, 3D model objektu (nielen jeho povrchu))
- Využitie z hľadiska účelu (podrobnejšie sa k účelu dostaneme priamo pri konkrétnom druhu signálu):
  - Kódovanie
    - transformačné kódovanie =
      - kompresia (zvyčajne sa myslí stratová)
      - kompakcia (zvyčajne sa myslí bezstratová)
  - parametre
    - kompresný pomer („compression ratio“)
    - kvalita (iba pri stratovom kódovaní)
      - objektívne vyhodnocovanie (napr. PSNR, MSE)
      - subjektívne vyhodnocovanie (MOS, ...)
  - analýza – typicky spektrálna analýza (frekvenčná analýza), časovo-frekvenčná analýza
  - úprava – typicky úprava frekvenčného spektra