

Obrazové Transformácie

Cvičenia z Počítačového Videnia

Zuzana Haladová
Júlia Kučerová

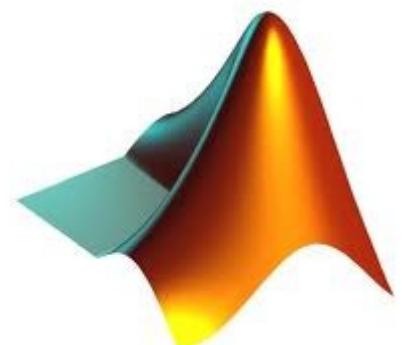
Obrazové transformácie

Zovšeobecnená Houghova transformácia

Rýchla Fourierova transformácia

Watershed segmentácia

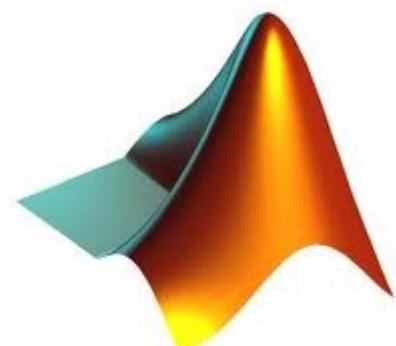
Úloha 2



Hough

Houghova transformácia- ak chceme detektovať oblasti zo známym tvarom hranice

- Vieme detektovať priamky aj krivky, ak je známe ich analytické vyjadrenie
- Metóda je robustná aj pre zašumené objekty



Hough

Houghova transformácia pre priamky

- Priamku vieme (v 2D) vyjadriť

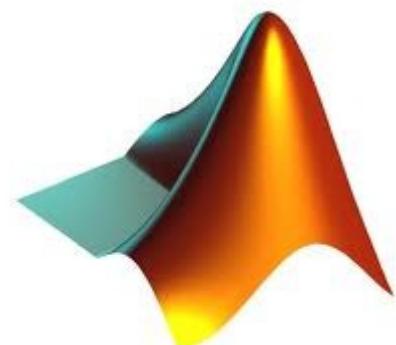
$$y = mx + b$$

- Houghova trans. reprezentuje priamku pomocou b , m resp. r , θ pričom

- r je najmenšia vzdialenosť od počiatku k priamke



- θ je uhol medzi x a priamkou



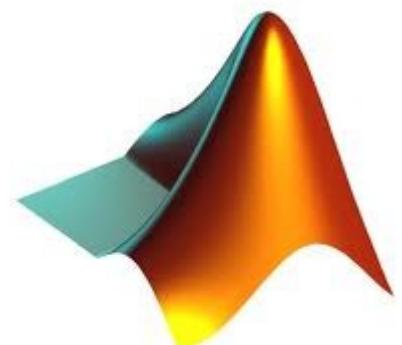
Hough

Vyjadrenie priamky pomocou r a θ je:

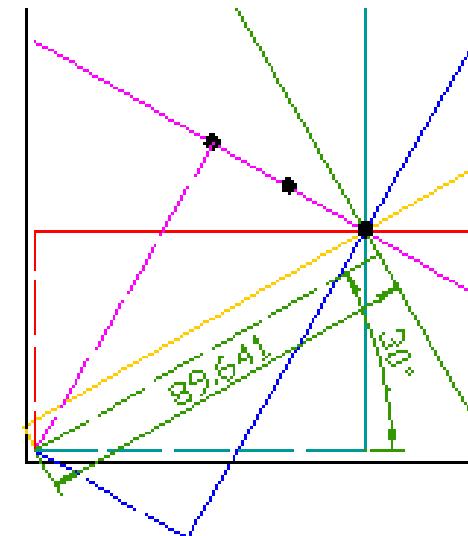
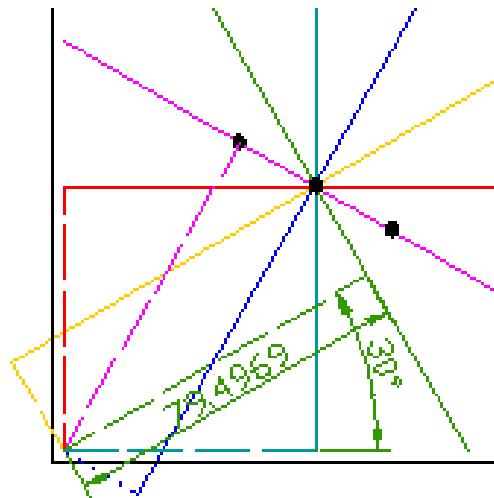
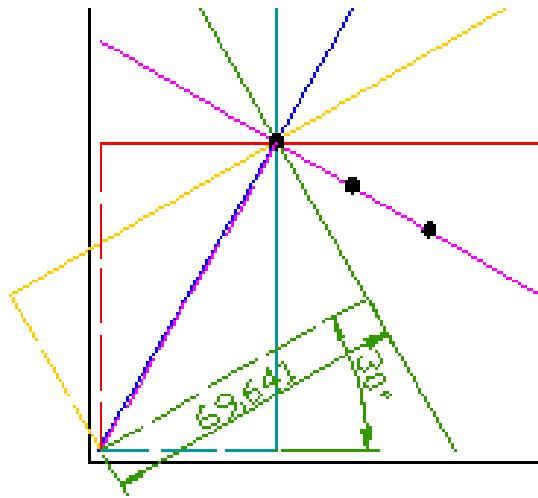
$$y = \left(-\frac{\cos \Theta}{\sin \Theta} \right) x + \left(\frac{r}{\sin \Theta} \right)$$

HT pracuje na bin. obrázku nasledovne:

- ❑ Pre každý bod obrázku a pre každú priamku ním prechádzajúcu vypočíta r a θ tejto priamky.



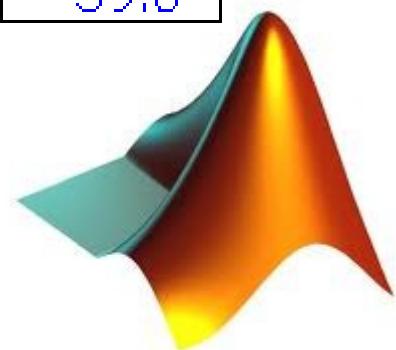
Hough



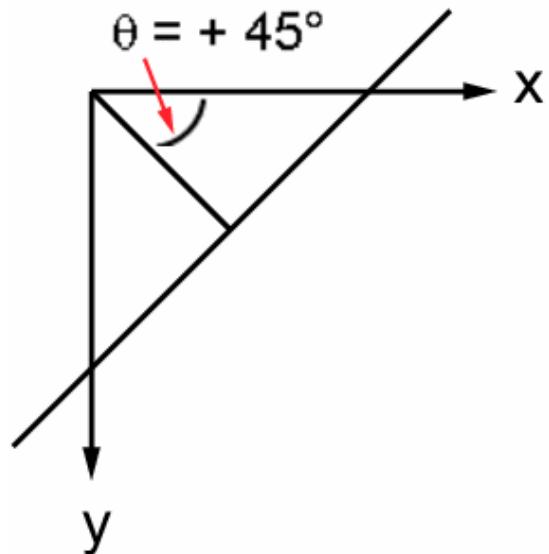
Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4

Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	-19.5

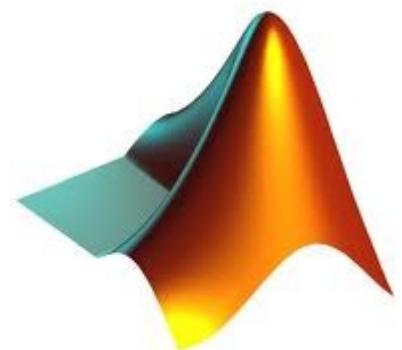
Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	-39.6



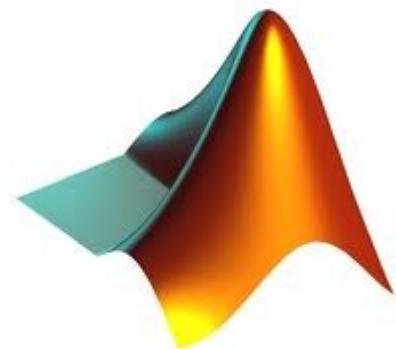
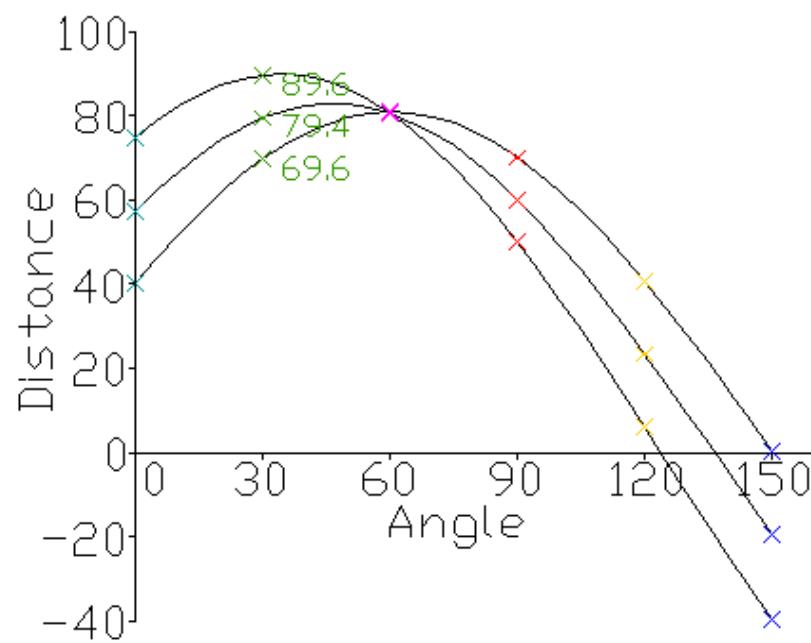
Hough



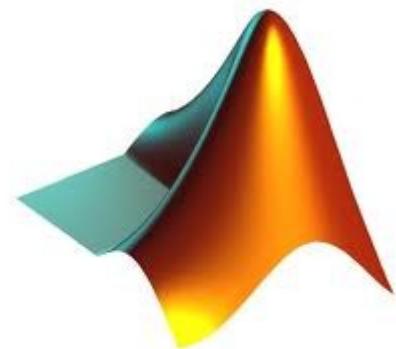
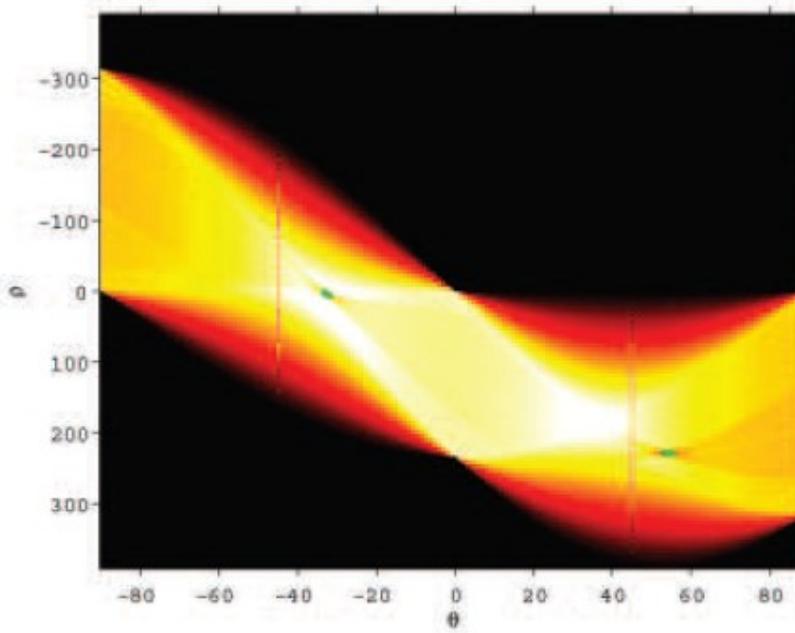
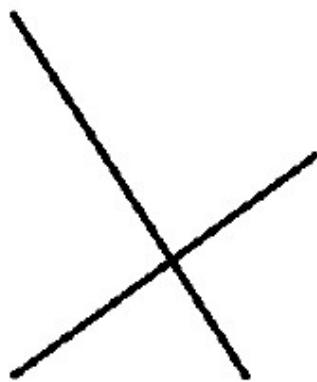
$$\rho = x \cos(\theta) + y \sin(\theta)$$



Hough



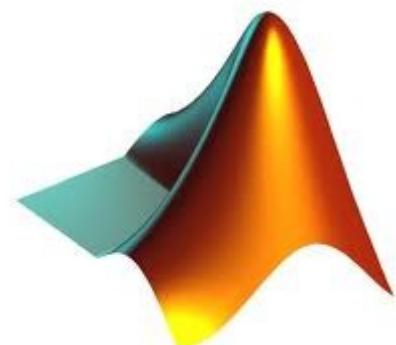
Hough



Hough

✓ MATLABE (Image Processing Toolbox)

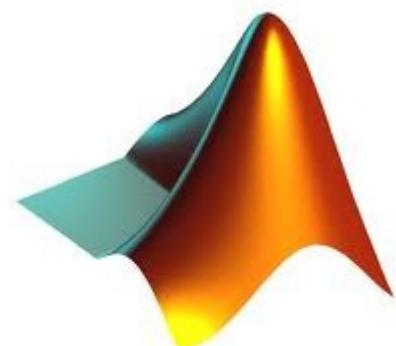
```
I=imread('1.jpg'); IG =rgb2gray(I);  
IE=edge(IG,'Canny');  
[H,T,R]=hough(IE);  
imshow(imadjust(mat2gray(H)),'XData',T,'YData',R);  
colormap(hot);
```



Hough

[H,T,R]=hough(I,E);

- ❑ H – matica Houghovej transformácie
- ❑ T je pole hodnôt θ a R je pole r nad ktorými je vyjadrená H
- ❑ Riadky a stĺpce H zodpovedajú r a θ
- ❑ hodnota na pozícii (1,2) zodpovedá počtu bodov ležiacich na priamke z 1. r a 2. θ z polí R a T



Hough

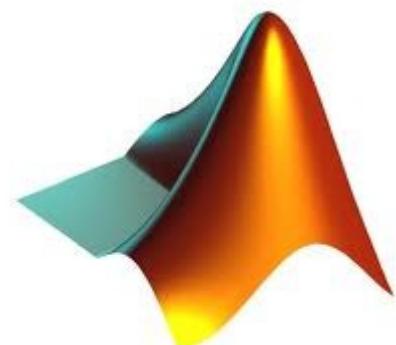
[H,T,R]=hough(I,E);

- peaks = houghpeaks(H, numpeaks)

- peaks je pole $Q \times 2$ kde Q

$<0, \text{numpeaks}>$

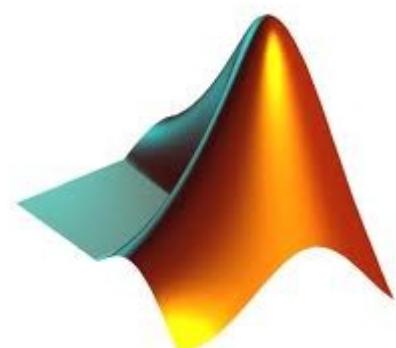
a obsahuje x,y hodnoty pre peaks v H



Hough

```
lines = houghlines(BW, theta, rho, peaks)
```

- ✉ lines je štruktúrované pole ktoré pre každú nájdenú úsečku obsahuje:
 - ✉ point1 x,y hodnoty konca úsečky
 - ✉ point2 x,y hodnoty konca úsečky
 - ✉ theta uhol v stupňoch (z matice H)
 - ✉ rho hodnota(z matice H)

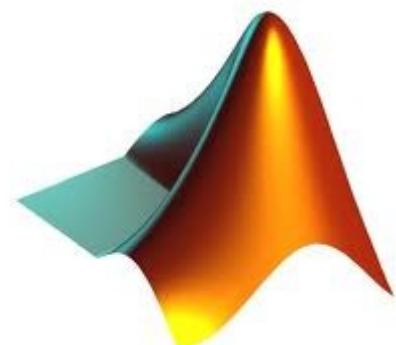


Hough

```
lines = houghlines(IE, theta, rho, peaks)
```

```
- figure, imshow(IE), hold on
```

```
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'Color','green');
end
```

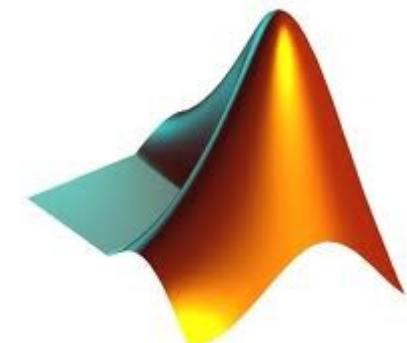
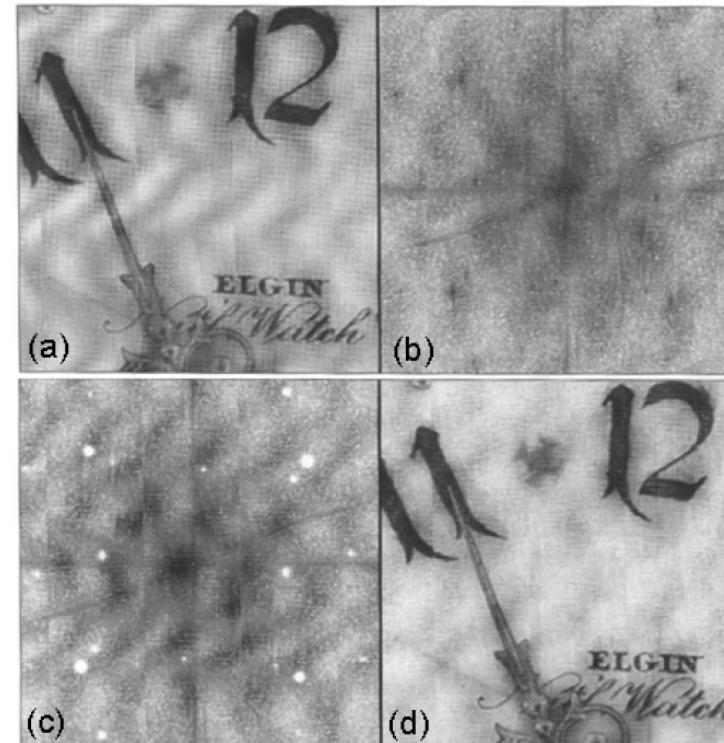


FFT

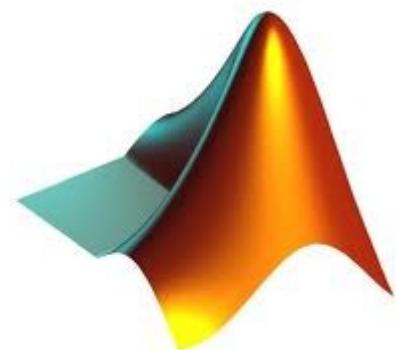
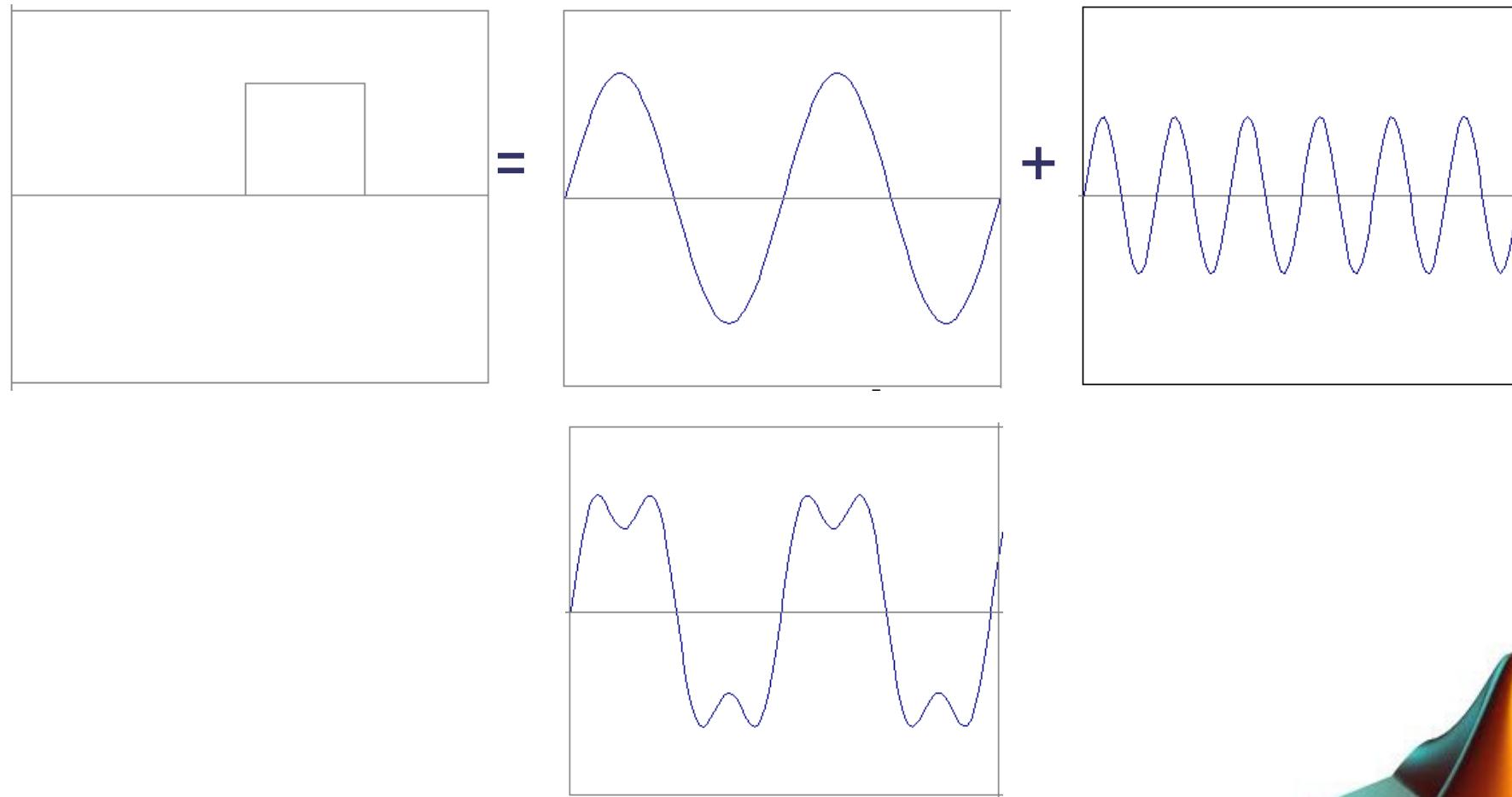
Akákoľvek funkcia $f(x)$ môže byť vyjadrená ako vážený súčet sínusov a kosínusov

Využitie:

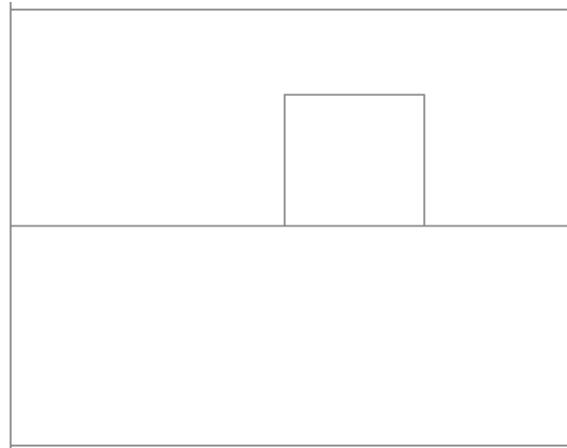
- lowpass filter
- highpass filter
- odstraňovanie artefaktov z obrazu



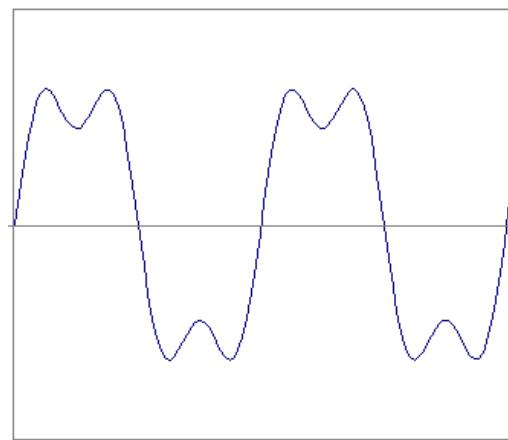
FFT



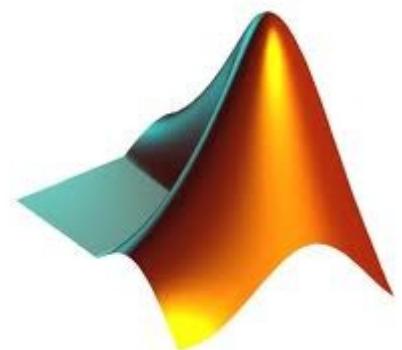
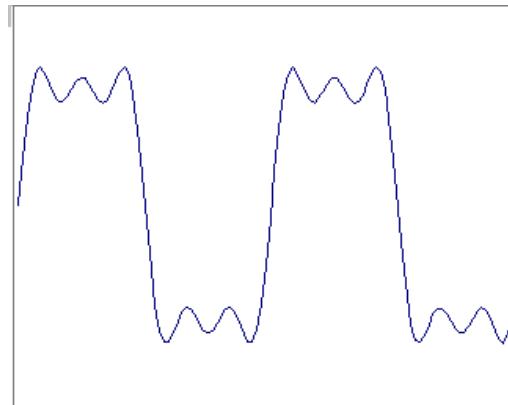
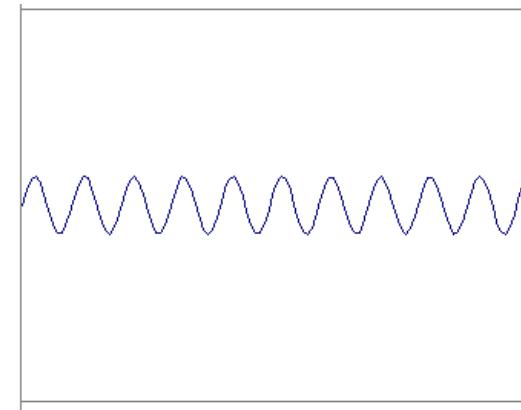
FFT



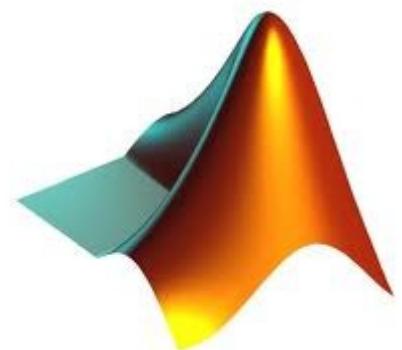
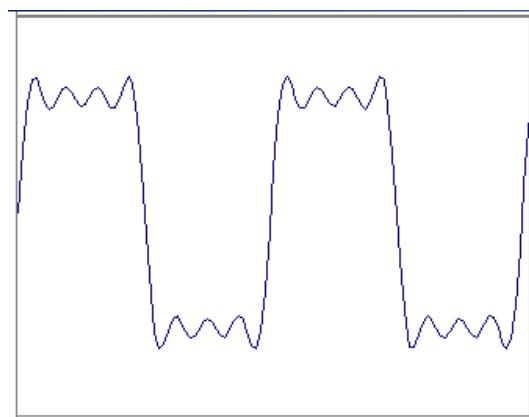
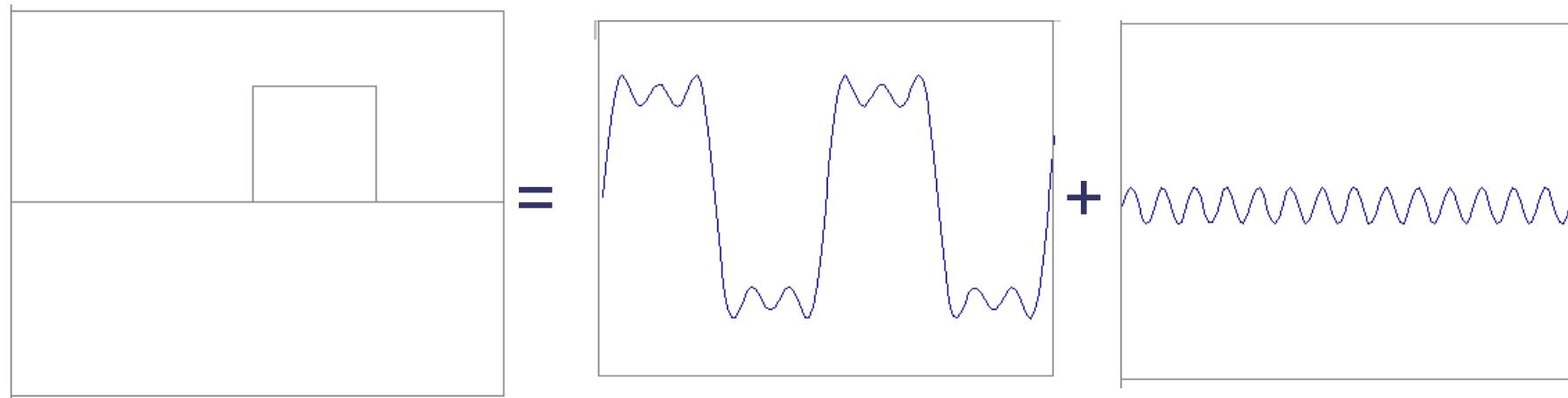
=



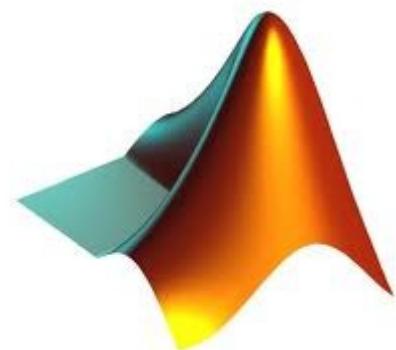
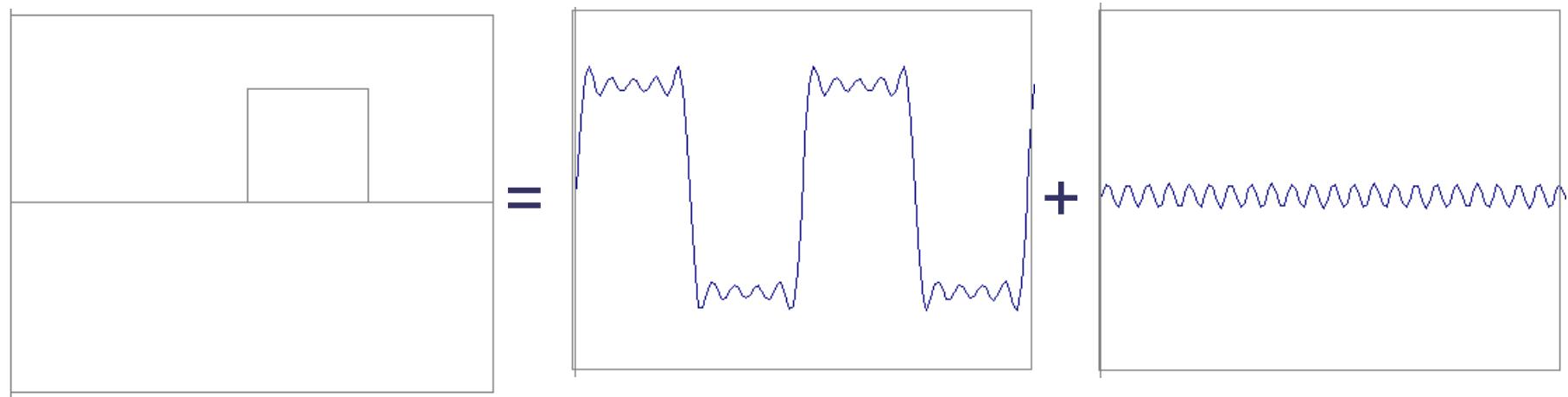
+



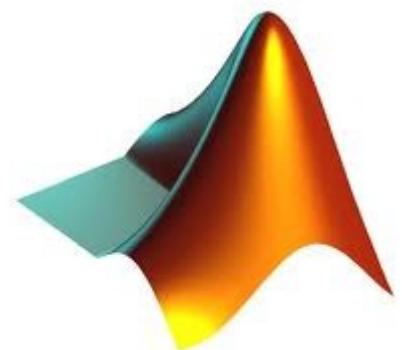
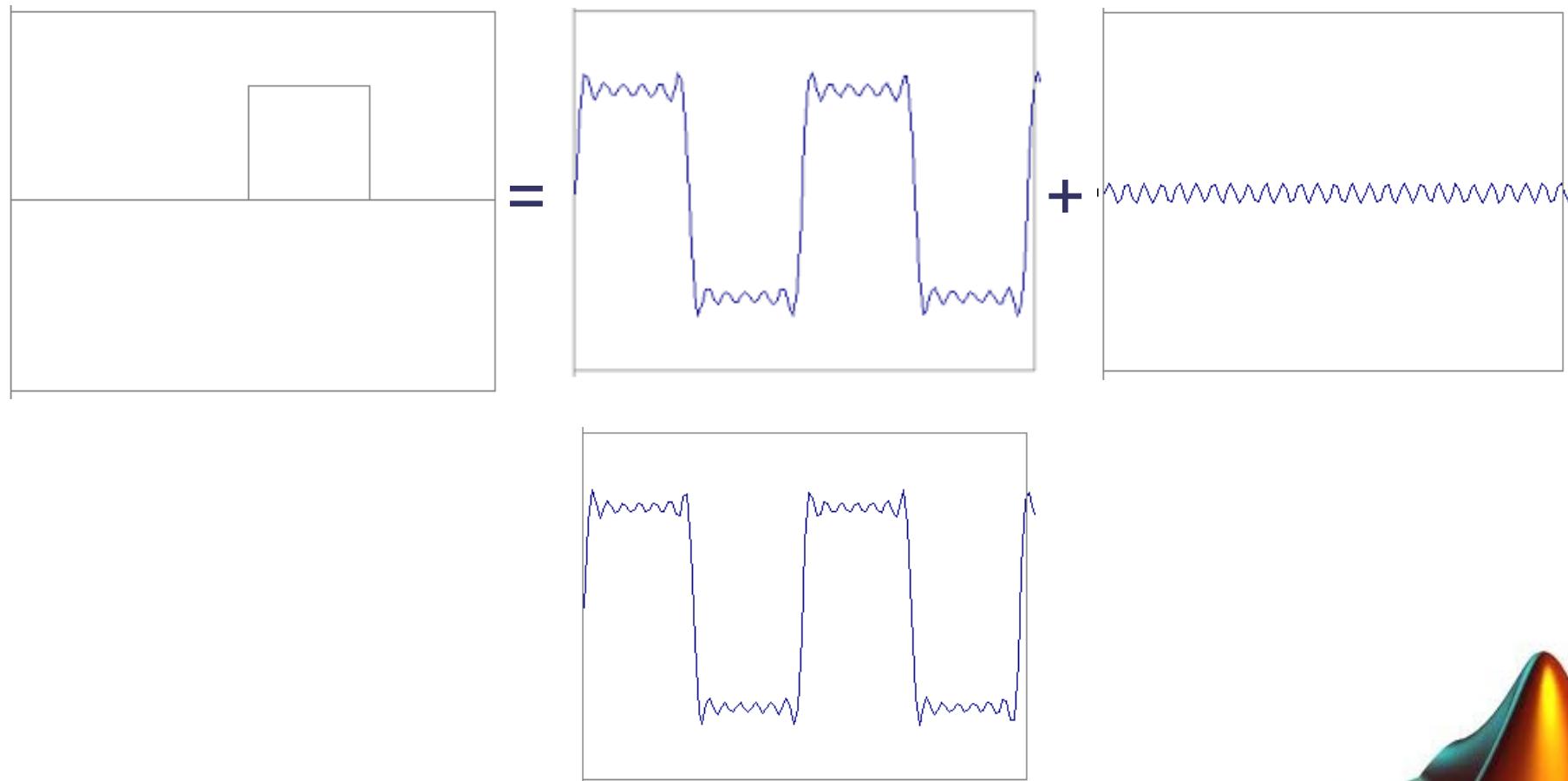
FFT



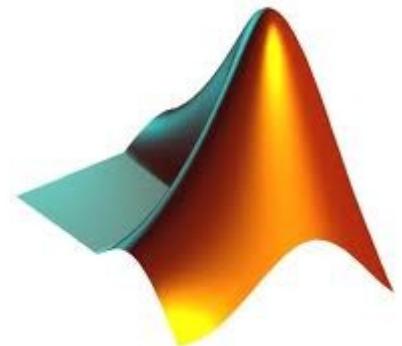
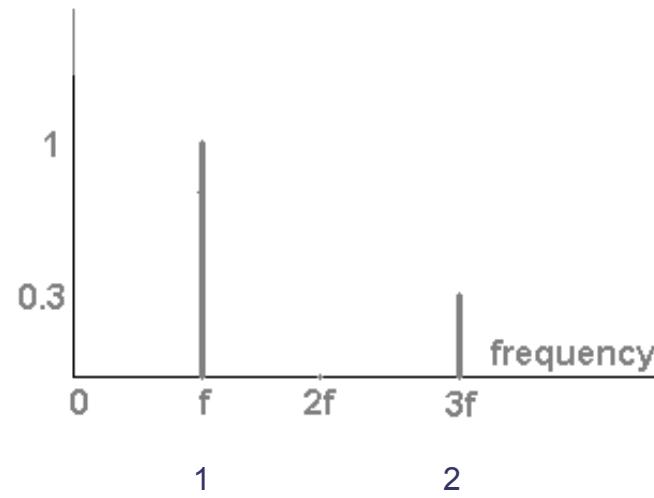
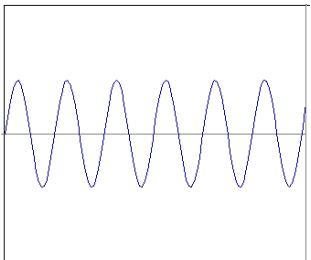
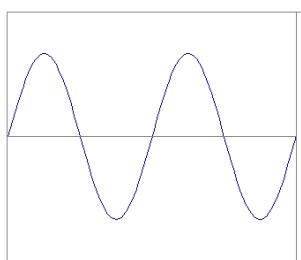
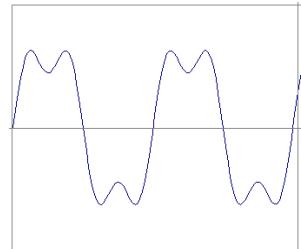
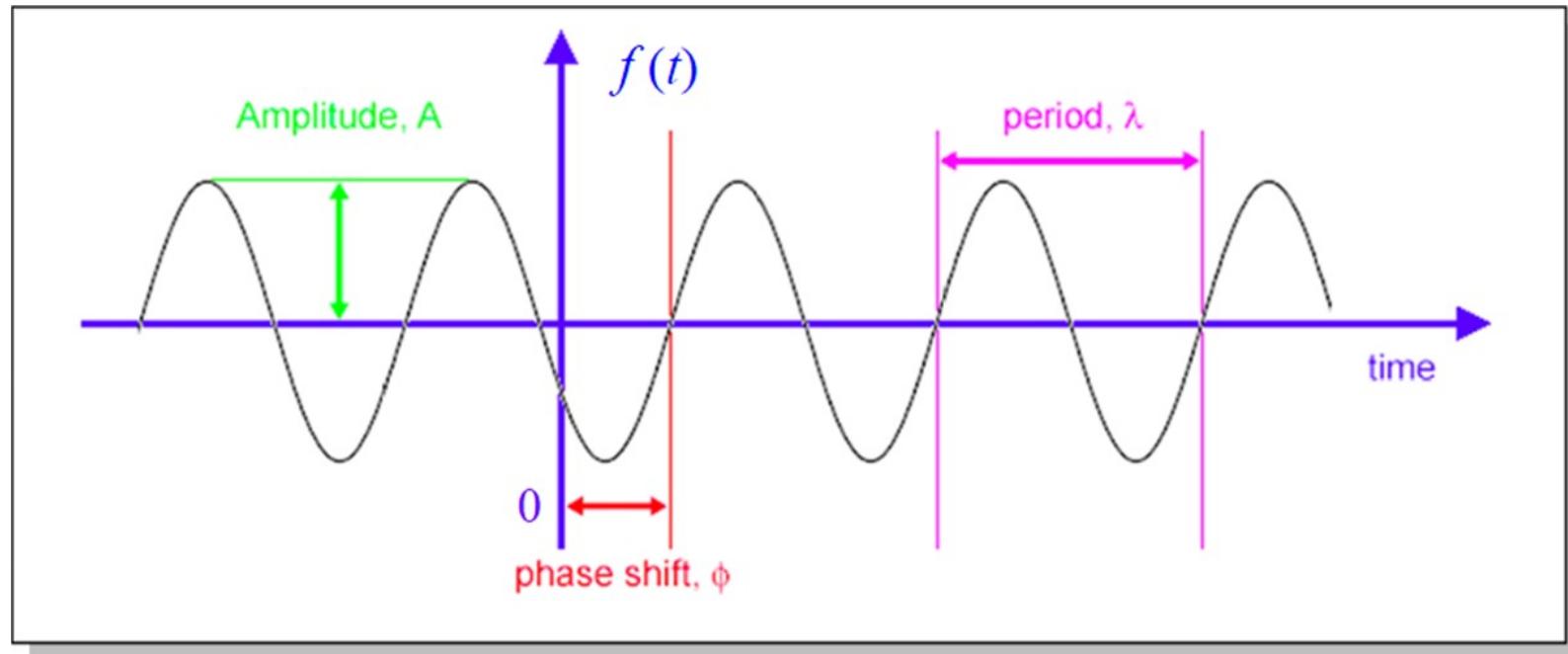
FFT



FFT

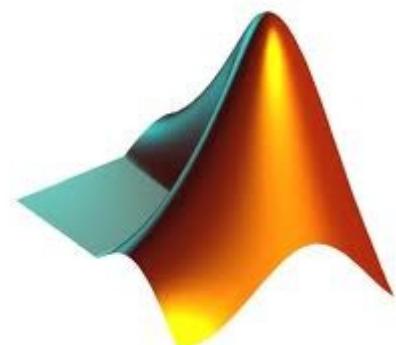


FFT



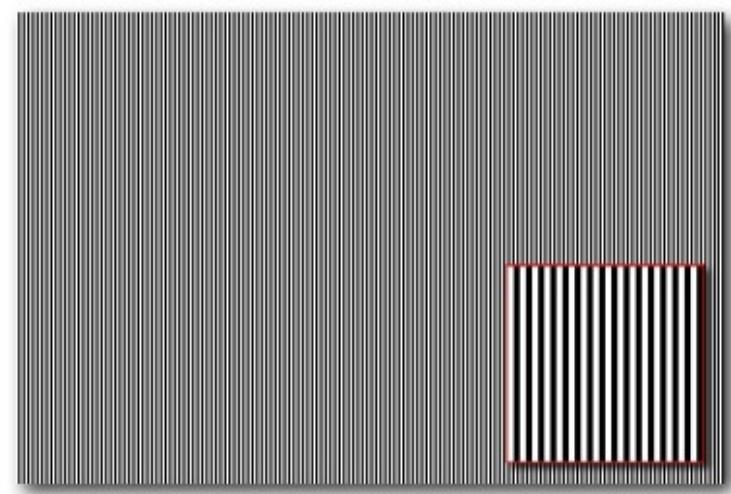
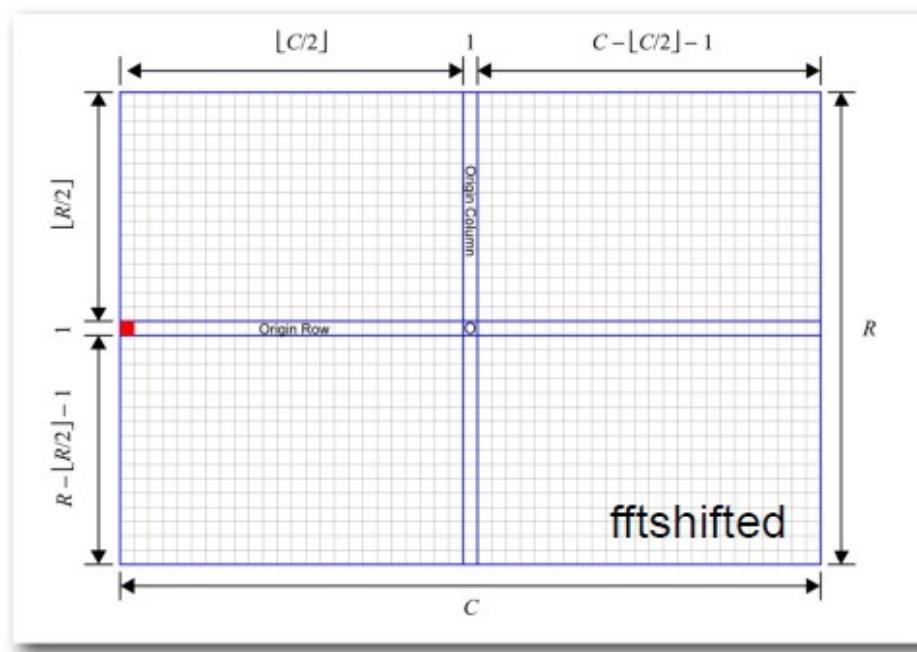
FFT - Skladanie signálu MATLAB

```
Fs = 1000; % Sampling frequency  
T = 1/Fs; % Sample time  
L = 1000; % Length of signal  
t = (0:L-1)*T; % Time vector  
  
y1 = 0.9*sin(2*pi*10*t);  
y2 = 0.3*sin(2*pi*30*t);  
y3 = y1 +y2;  
figure, plot(Fs*t(1:200),y1(1:200));  
figure, plot(Fs*t(1:200),y2(1:200));  
figure, plot(Fs*t(1:200),y3(1:200));
```

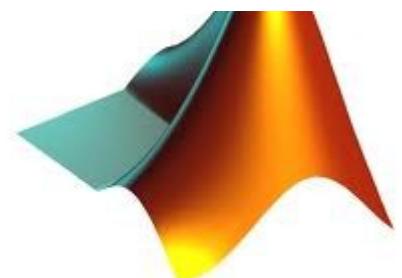


FFT

"horizontal" is the wavefront direction.

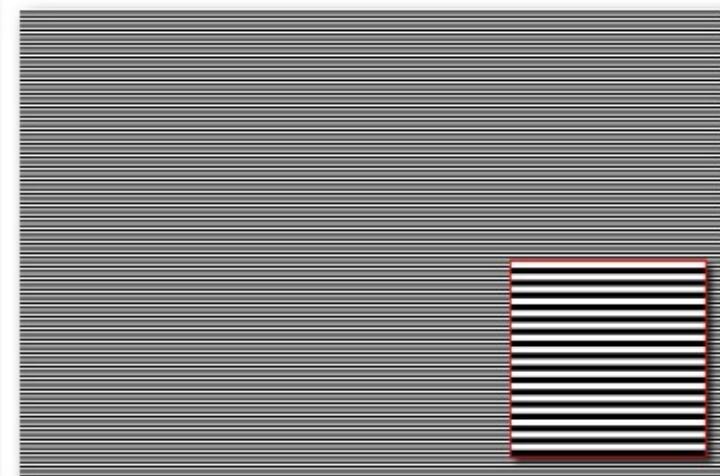
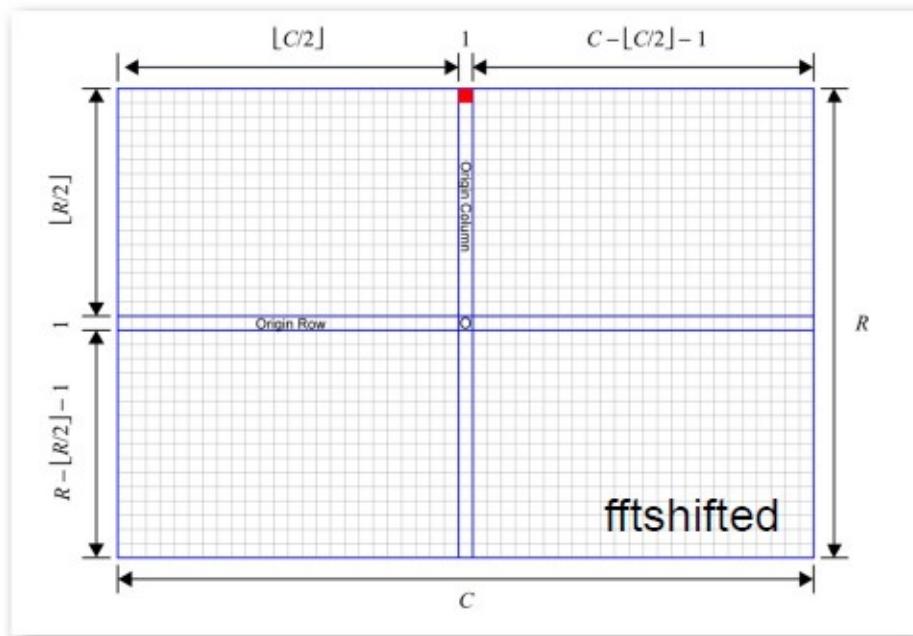


highest-possible-frequency horizontal sinusoid (C is even)

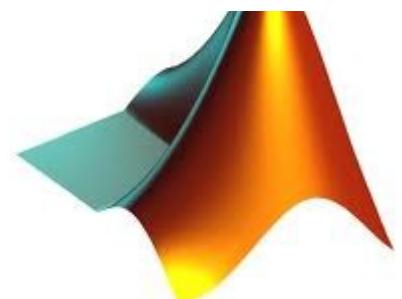


FFT

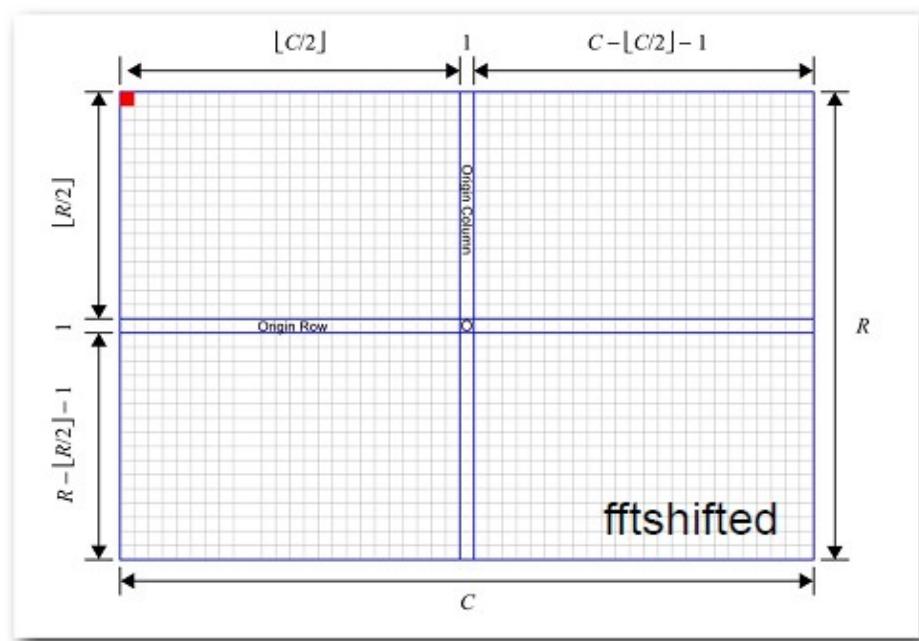
"vertical" is the wavefront direction.



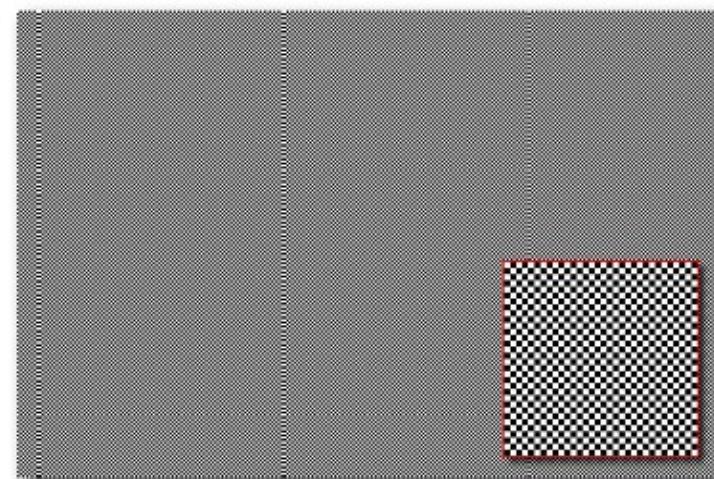
highest-possible-frequency vertical sinusoid (R is even)



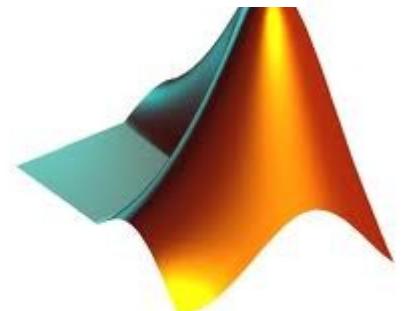
FFT



a checker-board pattern.

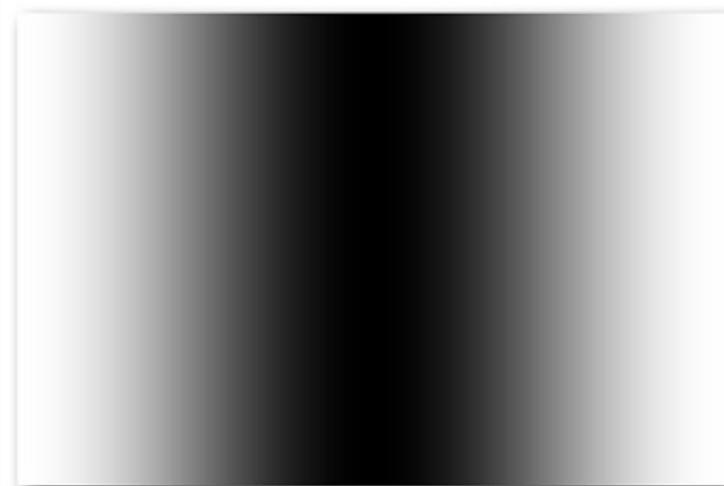
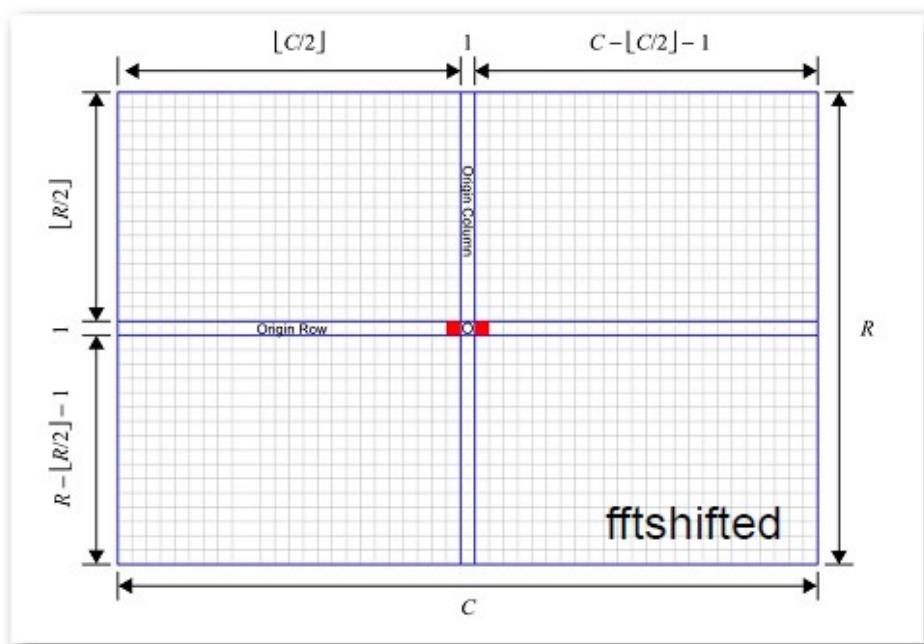


highest-possible-freq horizontal+vertical sinusoid (R & C even)

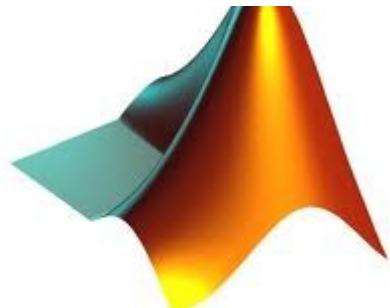


FFT

"horizontal" is the wavefront direction.

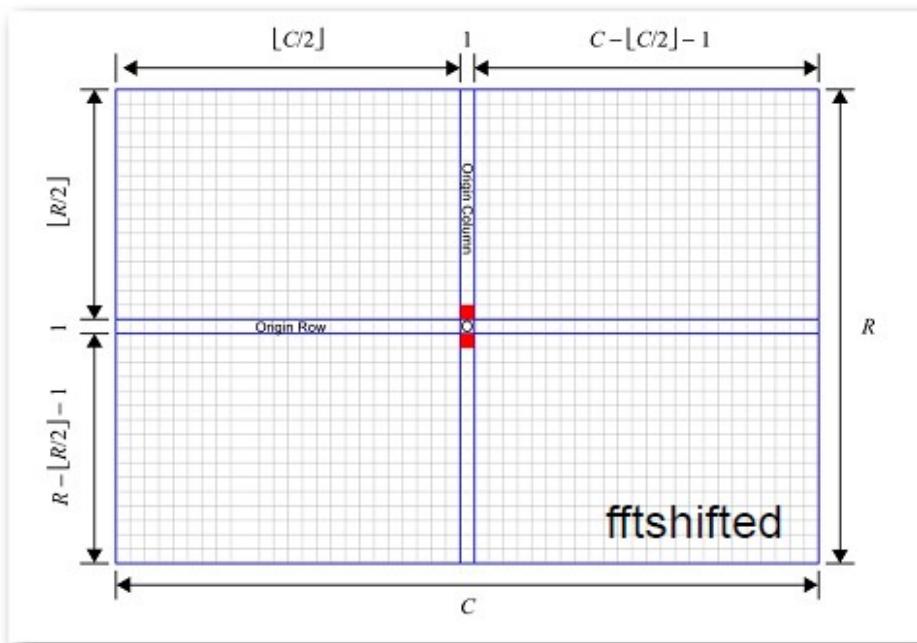


lowest-possible-frequency horizontal sinusoid

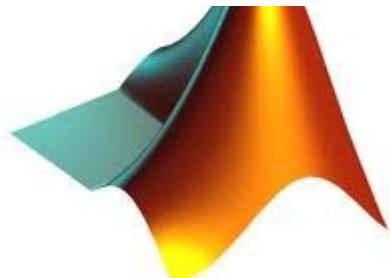


FFT

"vertical" is the wavefront direction.

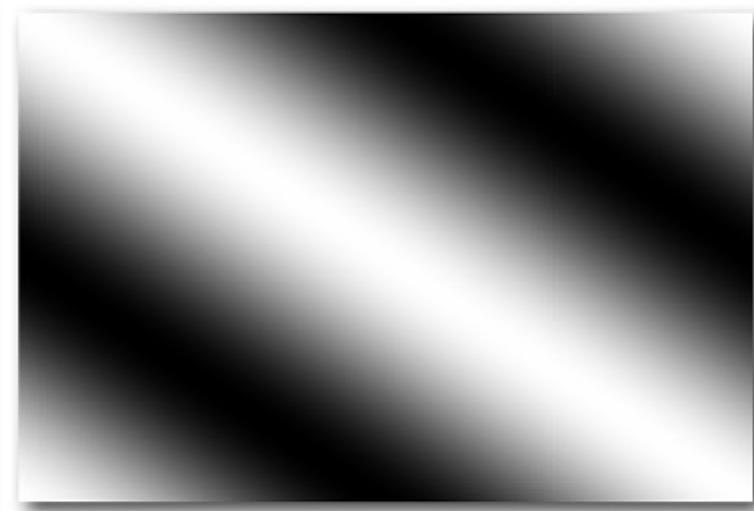
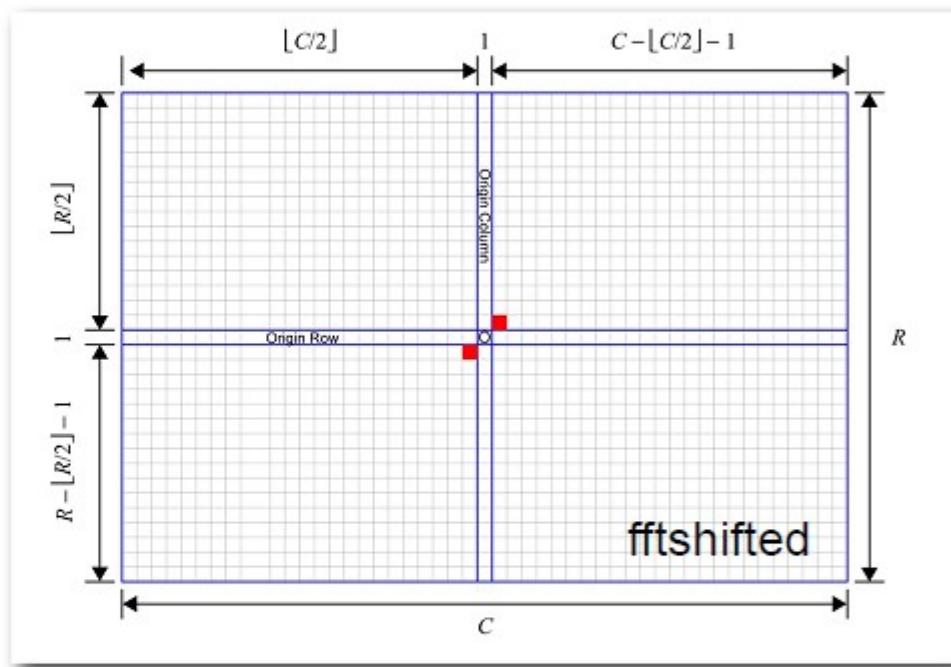


lowest-possible-frequency vertical sinusoid

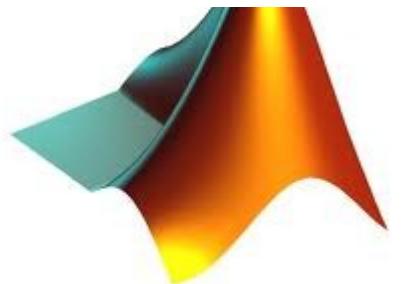


FFT

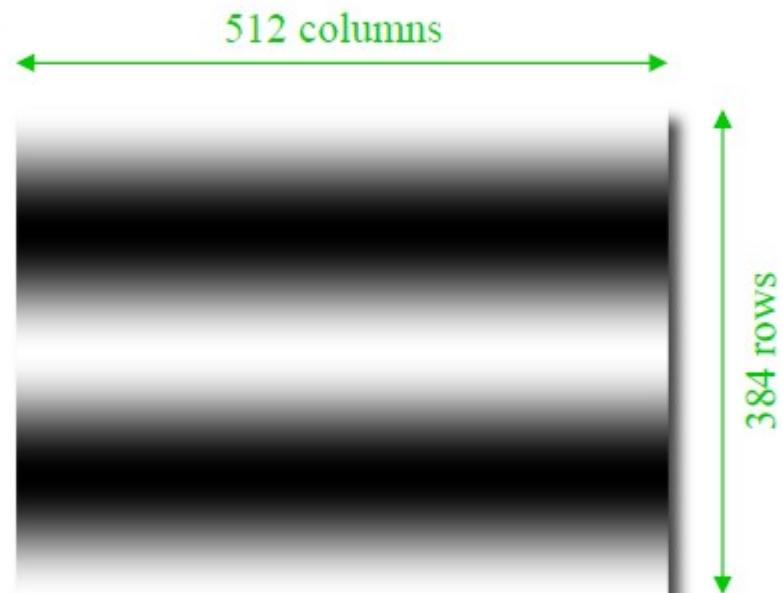
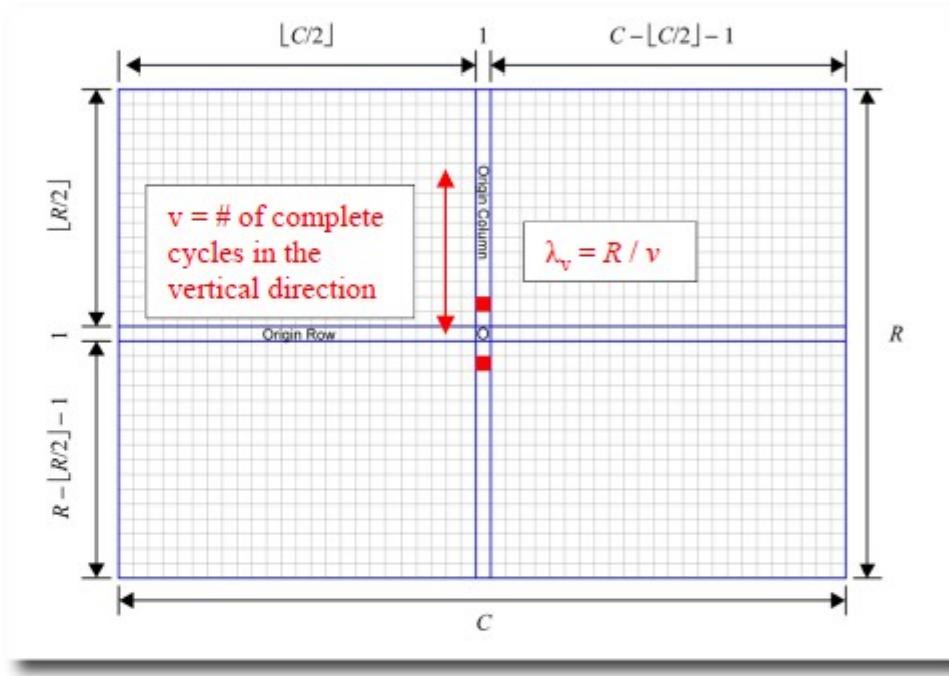
"negative diagonal" is
the wavefront direction.



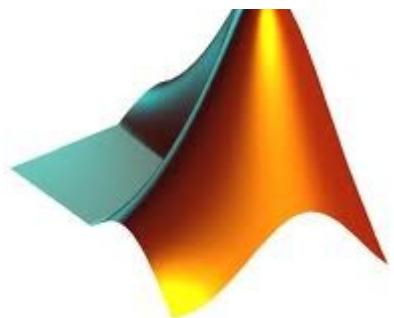
lowest-possible-frequency negative diagonal sinusoid



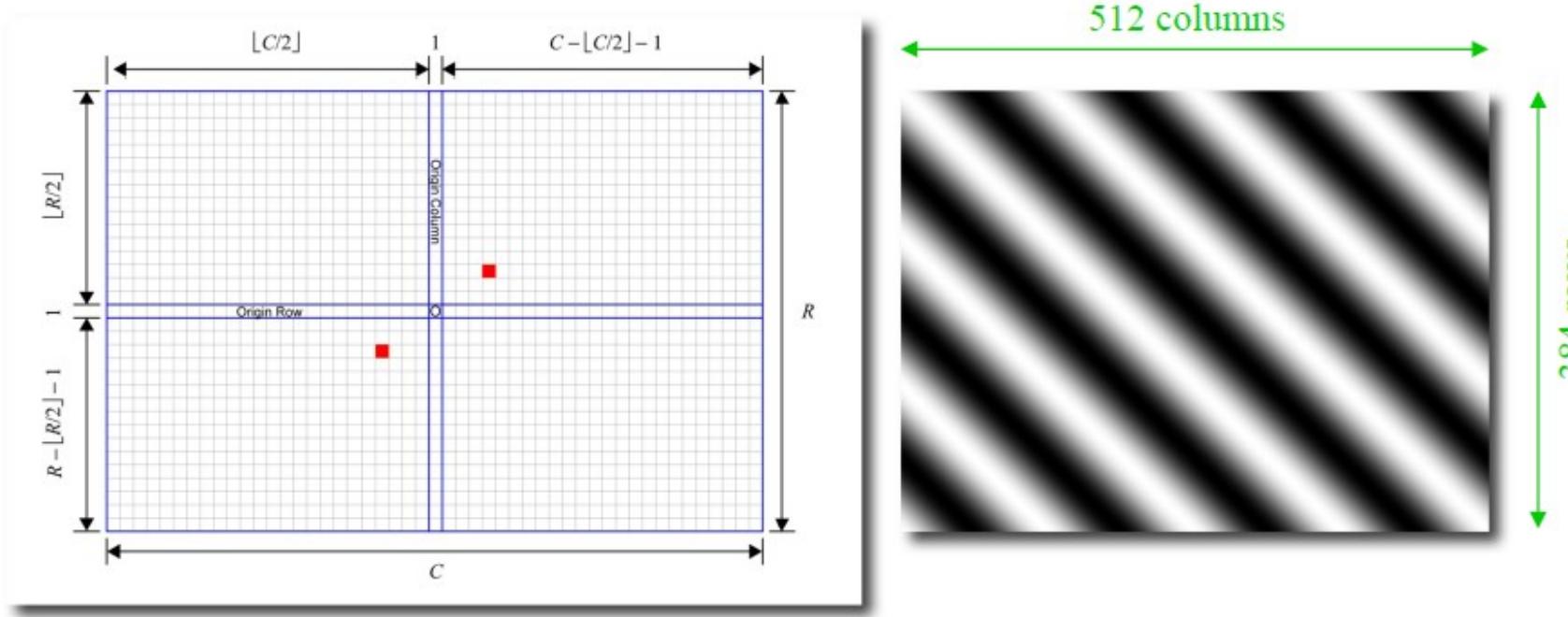
FFT



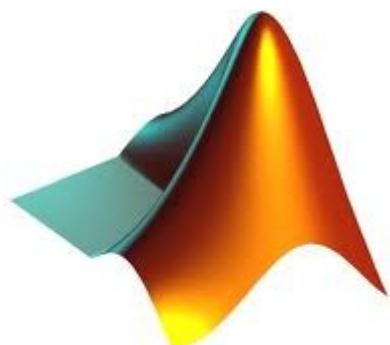
frequencies: $(u, v) = (0, 2)$; wavelength: $\lambda_v = 192$



FFT



frequencies: $(u, v) = (4, 3)$; wavelengths: $(\lambda_u, \lambda_v) = (128, 128)$



FFT

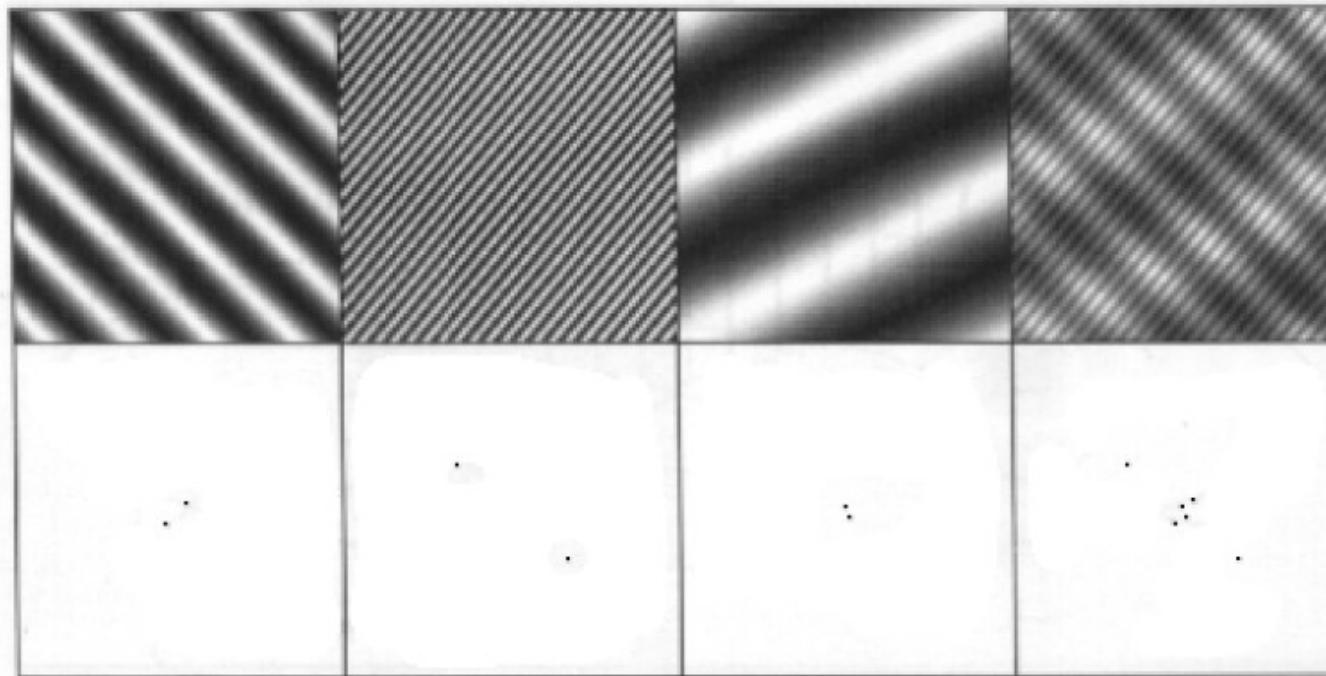
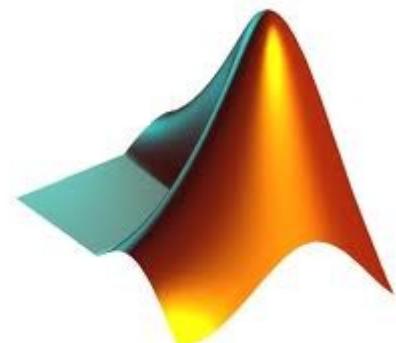
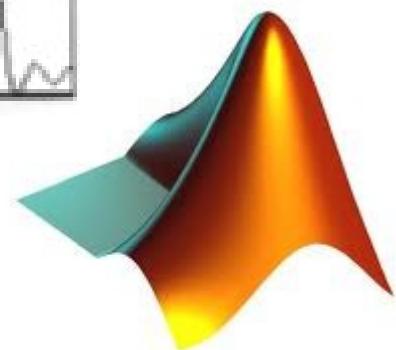
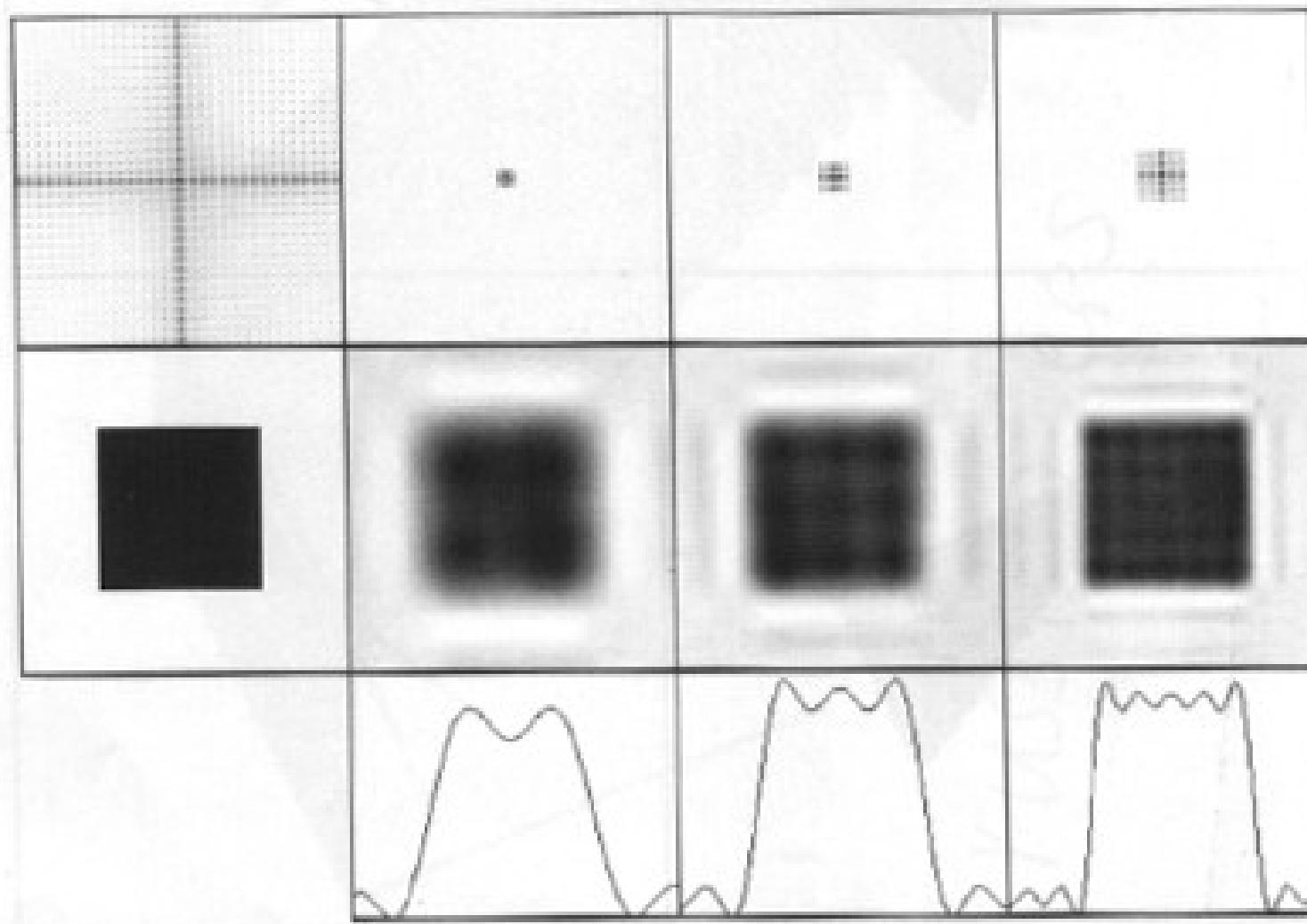


Figure 2: Images with perfectly sinusoidal variations in brightness: The first three images are represented by two dots. You can easily see that the position and orientation of those dots have something to do with what the original image looks like. The 4th image is the sum of the first three.

(Taken from p.177 of [1].)



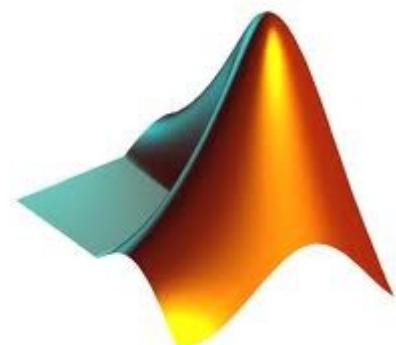
FFT



FFT MATLAB

2-D Fast Fourier transform

```
PS = fftshift(log(abs(fft2(I))+1));  
M = max(PS(:));  
imshow(uint8(255*(PS/M)));
```



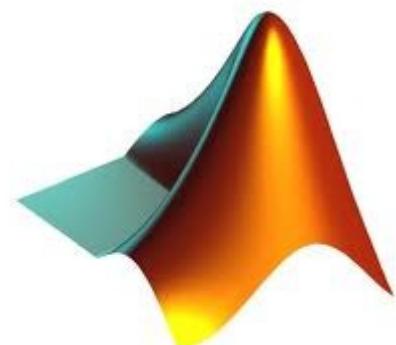
FFT ImageJ

- Image Processing and Analysis in Java
- Vytvorený v Java – umožňuje ho spustiť na všetkých OS
- Open Source

Fiji Is Just ImageJ

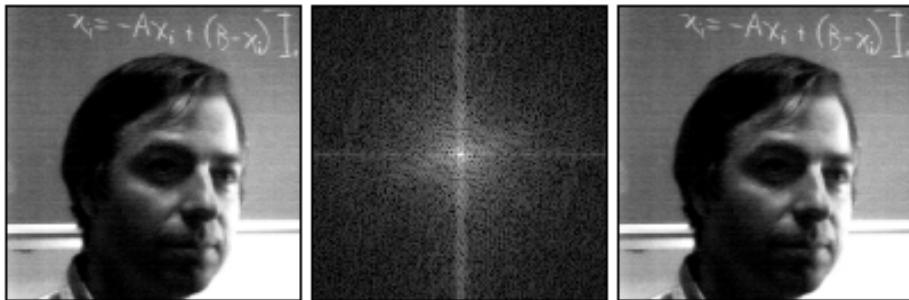
Fiji's main purpose is to provide a distribution of ImageJ with many bundled plugins

- <http://rsbweb.nih.gov/ij/>
- <http://fiji.sc/>

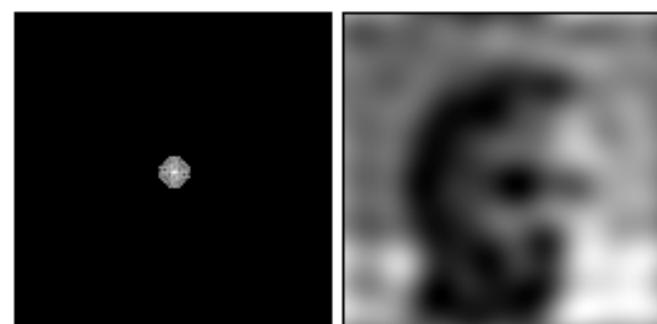


FFT ImageJ

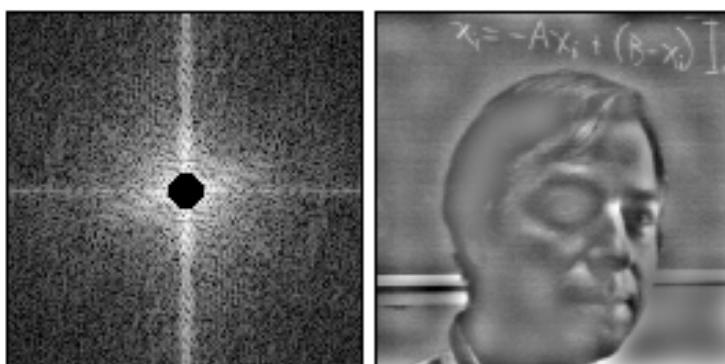
Brightness Image Fourier Transform Inverse Transformed



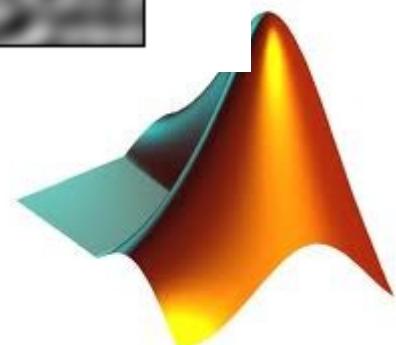
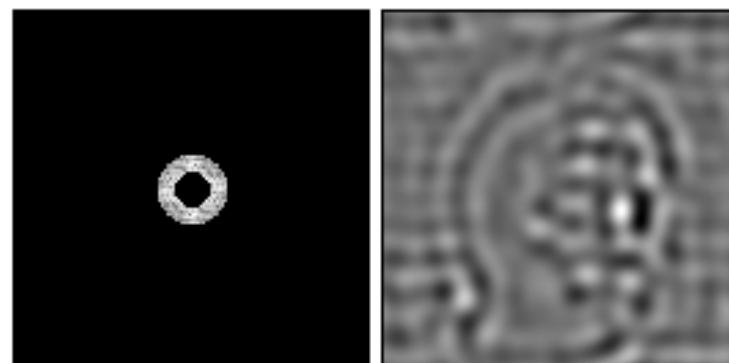
Low-Pass Filtered Inverse Transformed



High-Pass Filtered Inverse Transformed



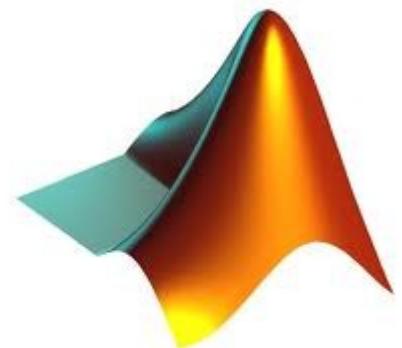
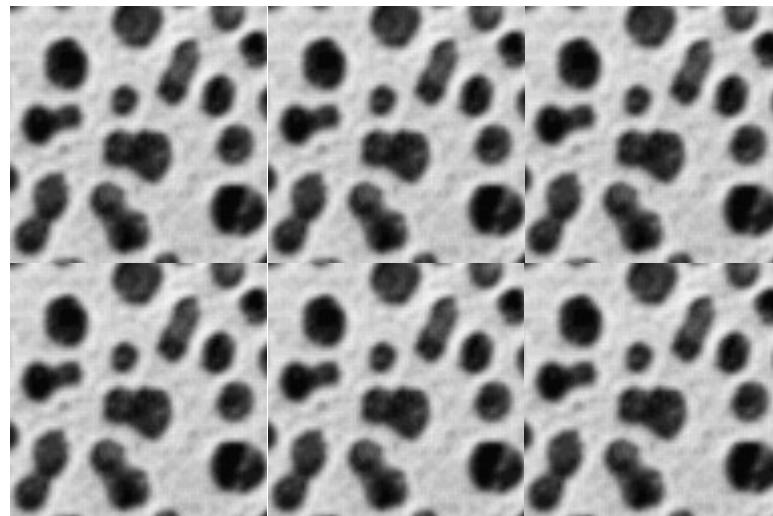
Band-Pass Filtered Inverse Transformed



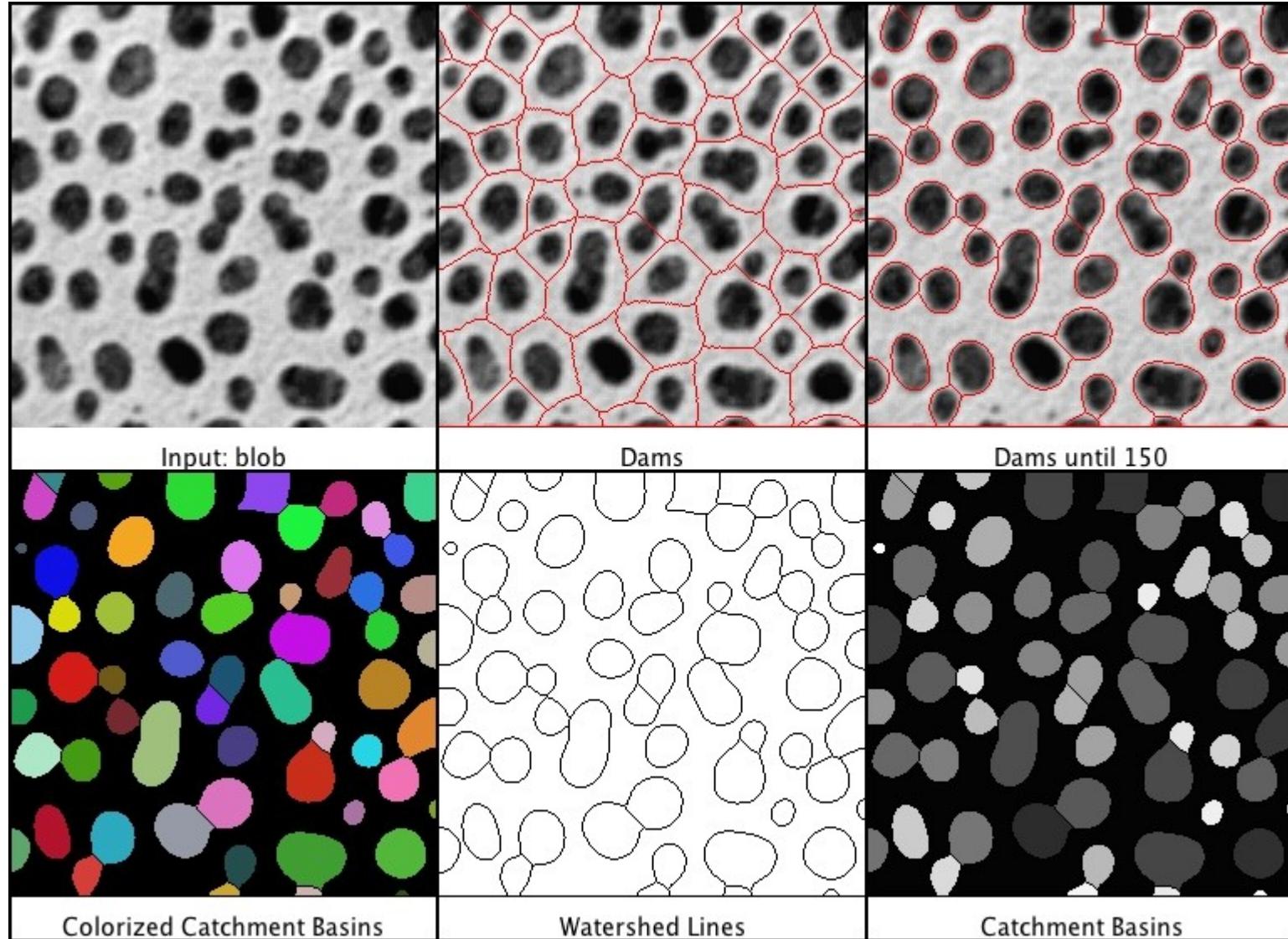
Watershed

Predstavíme si mimoúrovňový obrázok ako topograficky povrch

Začneme zaplavovať oblasti z lokálnych minim a zabránime spojeniu vody z rôznych zdrojov.

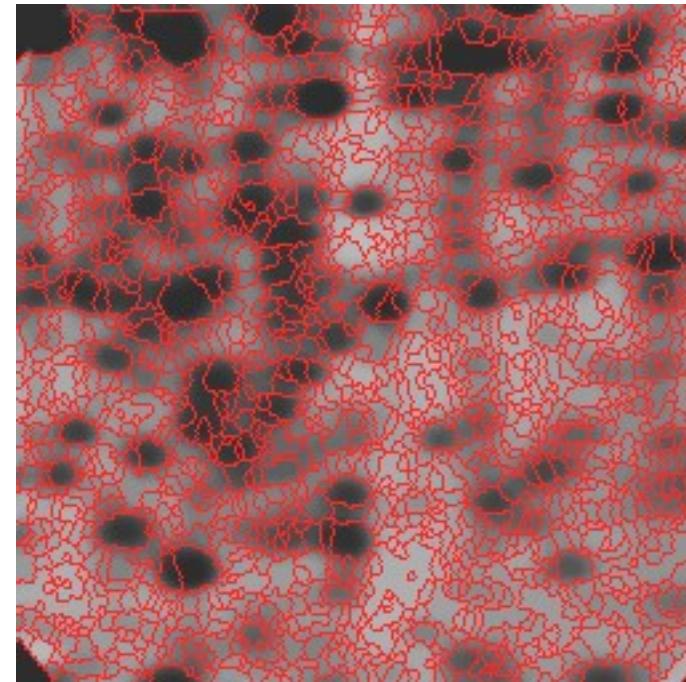
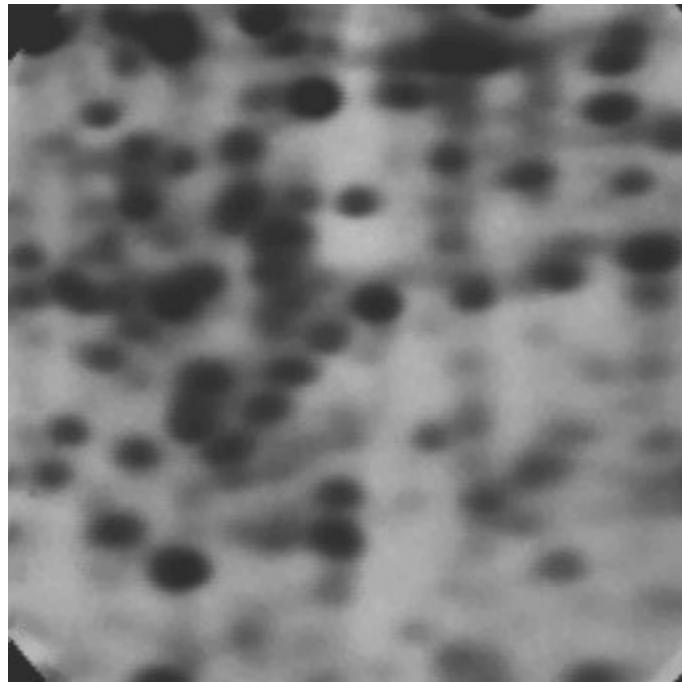


Watershed



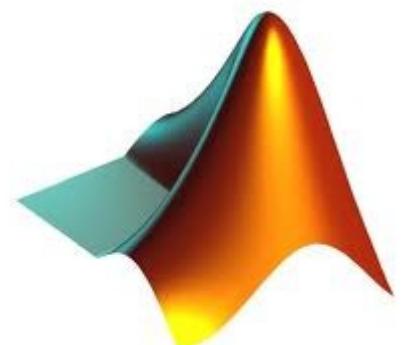
Watershed

Problém z metódou Watershed je nadsegmentácia



Watershed

- Watershed v IPT
- Šedoúrovňový, binárny, 3D...
- `I = imread('4.jpg');`
- `IG = togray(I);`
- `IW = watershed(IG); image(IW);`
- IW je matica `size(IG)` ktorá obsahuje číslo oblasti do ktorej daný pixel patrí



Uloha 2.

- Do 4.11.2012 do 23:59

Zadanie <http://www.sccg.sk/~haladova/Uloha2.html>

- Na DUvidenie@gmail.com
- subjekt DU2
- poslite priezvisko.zip obsahujuci:
 - Obrazok zo zadania
 - Vasu funkciu nazov.m

Uloha 2.

- Používajte:
- Odstraňovanie šumu
- Binarizácia obrázku
- Matematická morfologia
- Oddelenie objektov
- Popis vlastností objektov