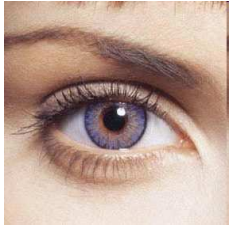


Farebné videnie s Octave

Seminár z UI

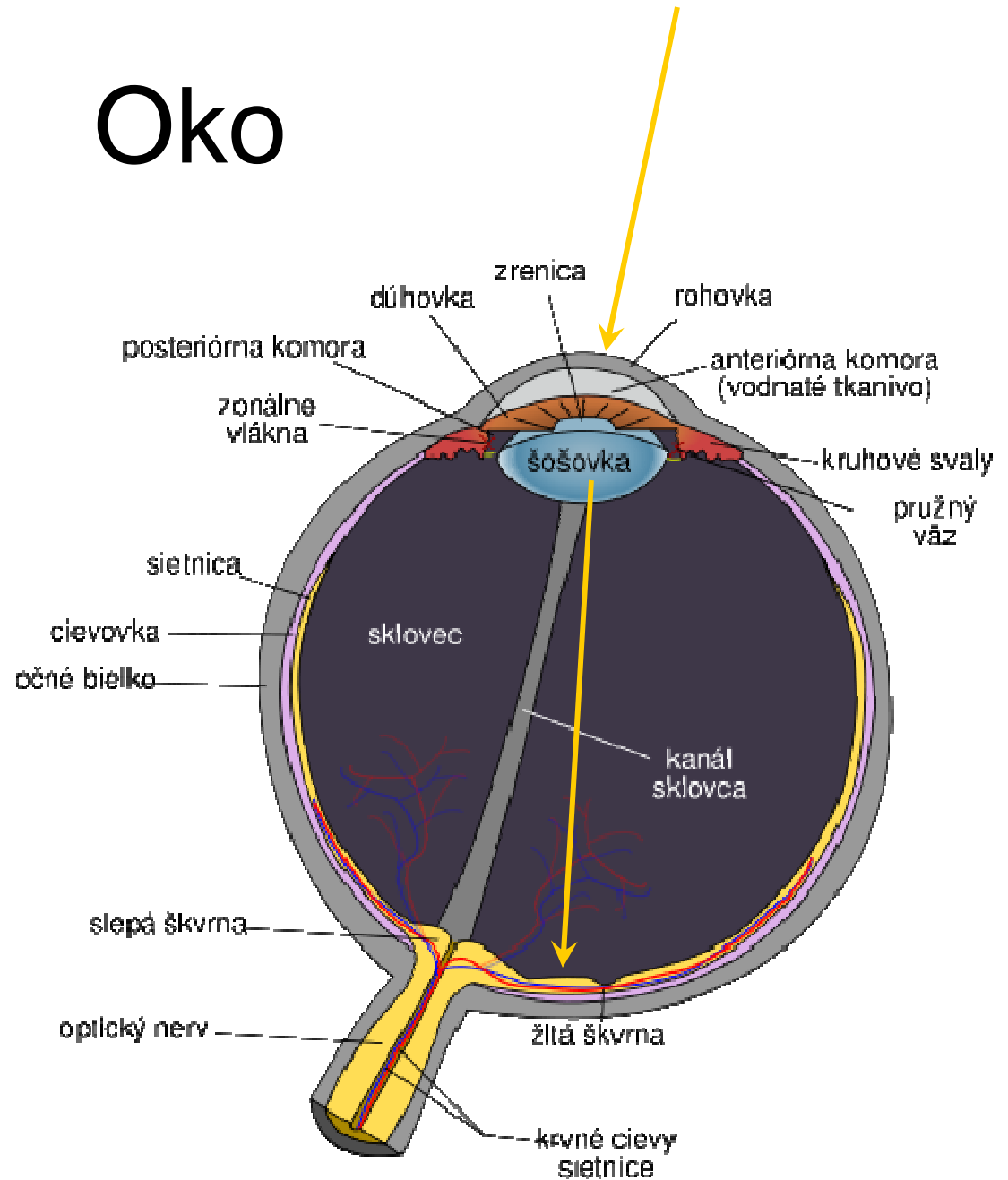
Andrej Lúčny

www.microstep-mis.com/~andy



Oko

- Optické zariadenie, ktoré zaručí, že na určitý bod sietnice dopadne svetlo prichádzajúce z jedného smeru.



Farby

- V každom okamihu dopadá na každý bod sietnice množstvo fotónov rôznej vlnovej dĺžky. Meranie na jednom bode sietnice za čas Δt dá celú distribúciu početnosti fotónov určitej vlnovej dĺžky

fotons.m

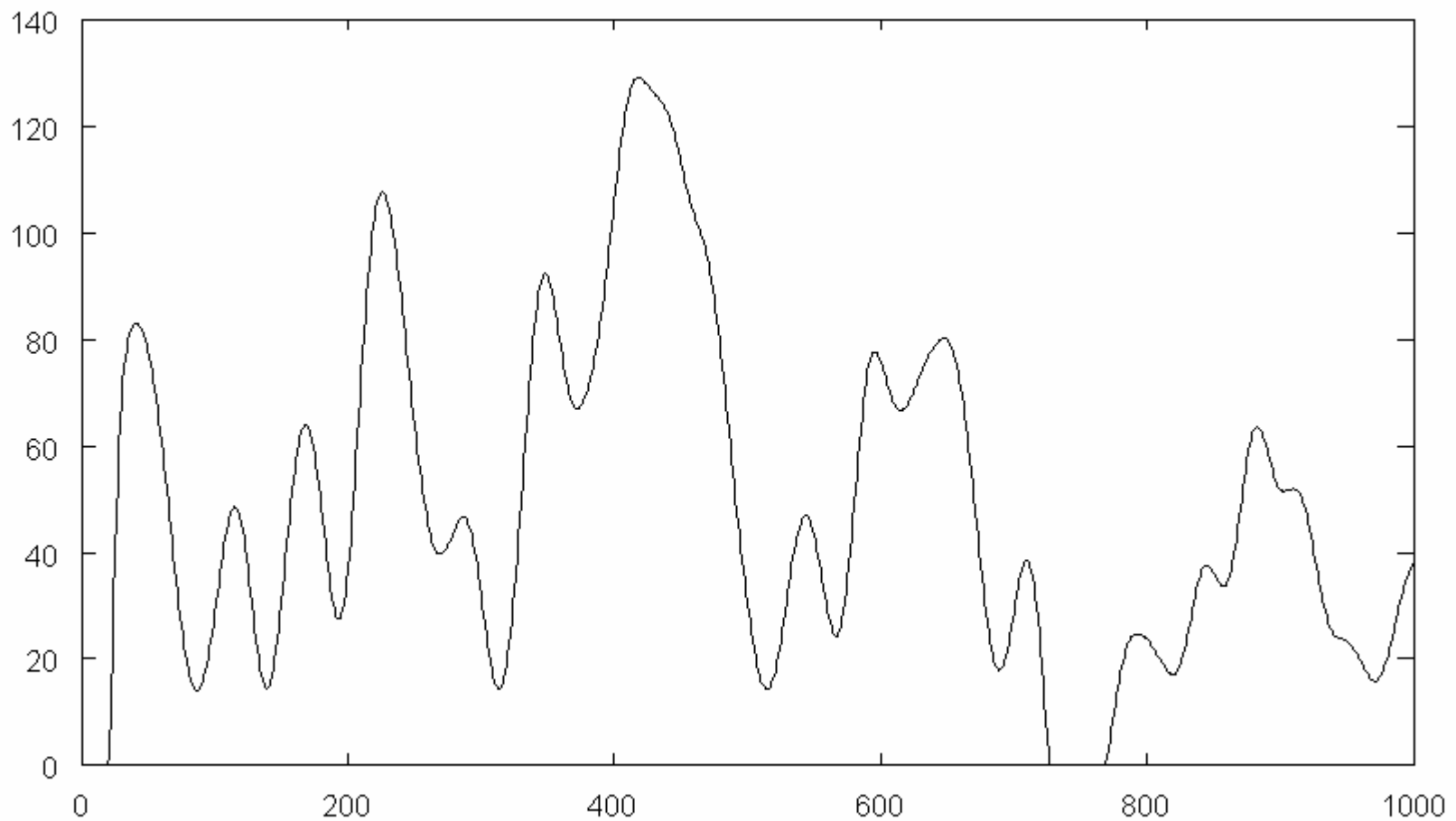
```
n = 1100;
pole = unifrnd(0,70,1,n);
plot(pole);
pause;
m = length(pole);
pole = iterate(pole,4,10);
m = 1000;
pole = pole(1:m);
pole = max(pole.-30, repmat(0,1,m));
pole = pole .* 10;
plot(pole);
```

iterate.m

```
function res = iterate(pole,iter,w)
m=length(pole);
for i=1:iter
    nove = zeros(1,m);
    for j=w+1:m-w
        nove(j)=mean(pole(j-w:j+w));
    end
    pole = nove;
end
res = pole;
endfunction
```

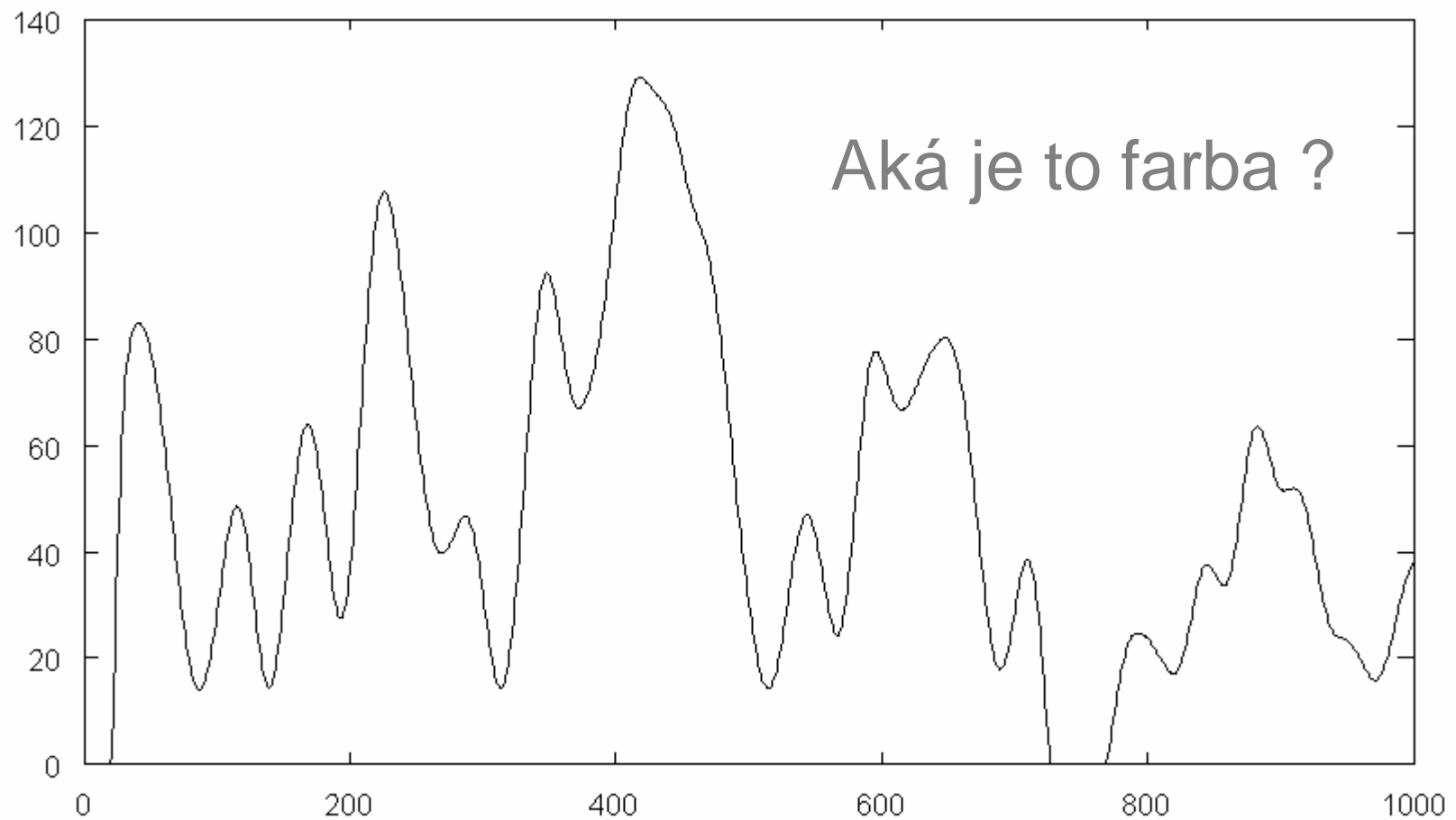
Farby

Príklad signálu - počty fotónov (v zmyslenej jednotke) za jednotku času



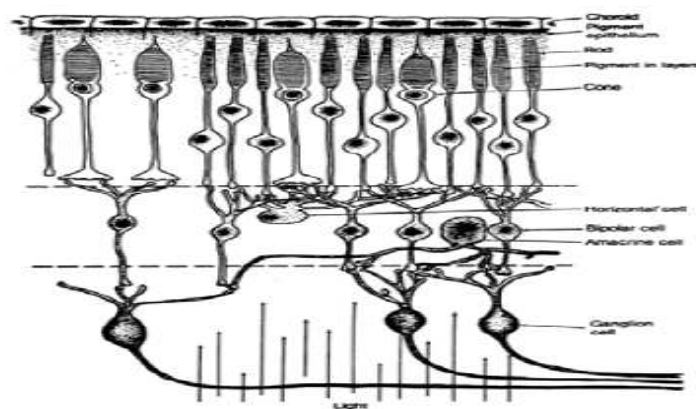
Farby

Príklad signálu - počty fotónov (v zmyslenej jednotke) za jednotku času



Farby

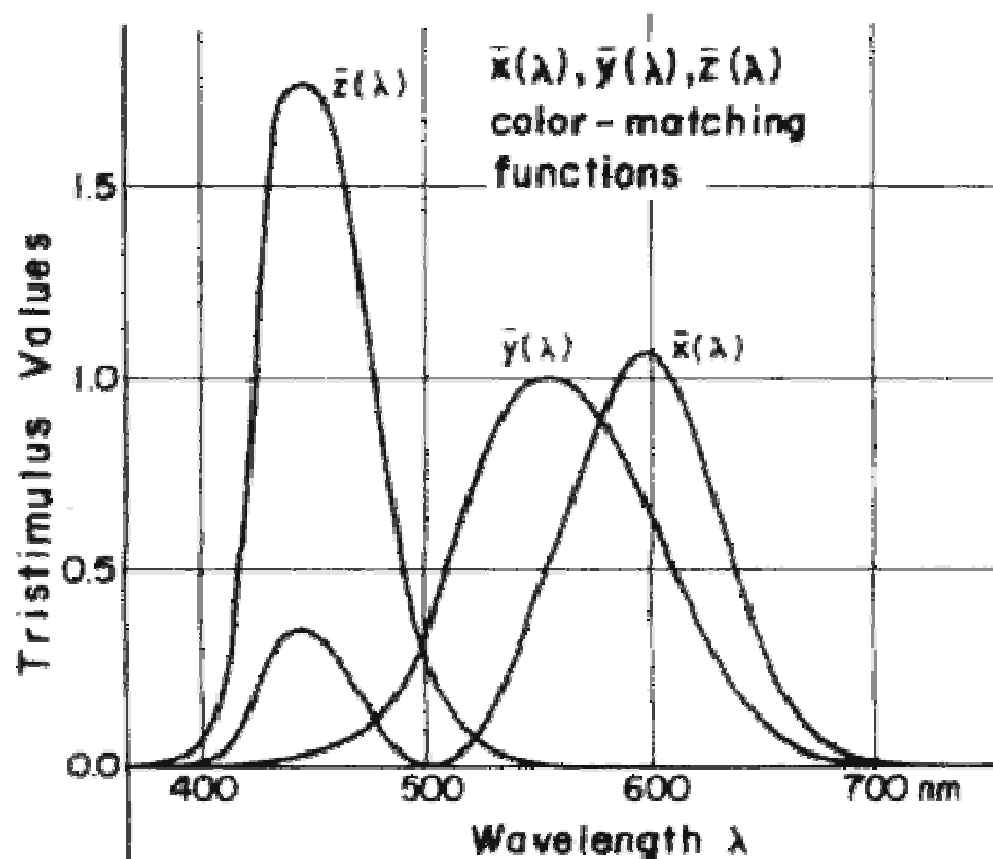
Na sietnici oka sú v jednom „bode“ buňky obsahujúce bielkovinový receptor – čapíky - slúžiace na farebné videnie. Príslušný receptor mení svoju konformáciu vplyvom dopadu fotónov určitej vlnovej dĺžky a vyvoláva signál tomu zodpovedajúcej intenzity idúci do mozgu.



Farby

V ľudskom oku sú tri druhy čapíkov, pričom každý inak reaguje na svetlo určitej vlnovej dĺžky

(X, Y, Z) - System



CIE 1931

loadcmf.m

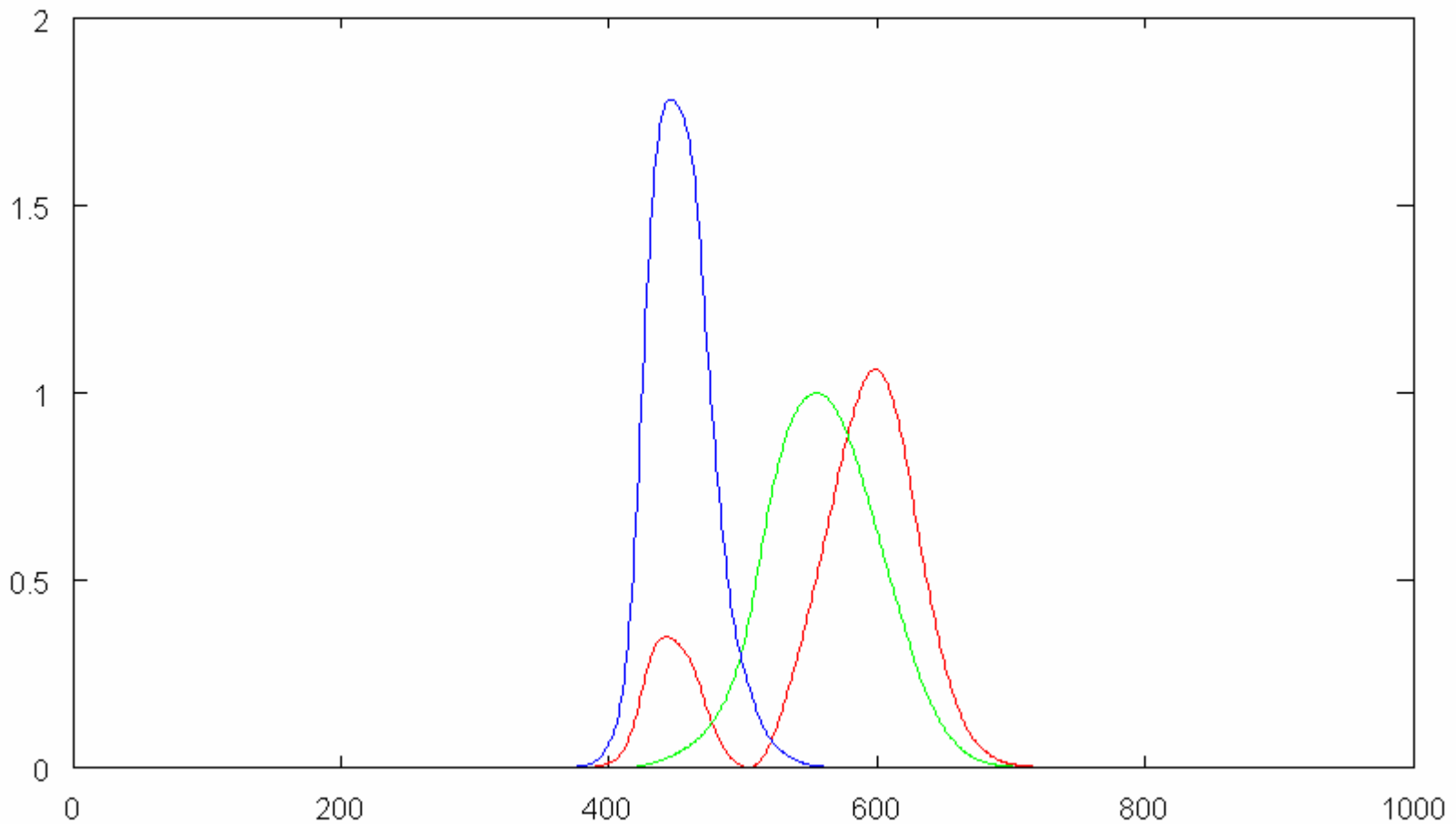
```
cmf = [  
    360, 0.000129900000, 0.0000003917000, 0.000606100000;  
    361, 0.000145847000, 0.0000004393581, 0.000680879200;  
    362, 0.000163802100, 0.0000004929604, 0.000765145600;  
    363, 0.000184003700, 0.0000005532136, 0.000860012400;  
    364, 0.000206690200, 0.0000006208245, 0.000966592800;  
    365, 0.000232100000, 0.0000006965000, 0.001086000000;  
    366, 0.000260728000, 0.0000007813219, 0.001220586000;  
    367, 0.000293075000, 0.0000008767336, 0.001372729000;  
    368, 0.000329388000, 0.0000009839844, 0.001543579000;  
    369, 0.000369914000, 0.0000011043230, 0.001734286000;  
    370, 0.000414900000, 0.0000012390000, 0.001946000000;  
    371, 0.000464158700, 0.0000013886410, 0.002177777000;  
    372, 0.000518986000, 0.0000015557280, 0.002435809000;  
    373, 0.000581854000, 0.0000017442960, 0.002731953000;  
    ...  
    830, 0.000001251141, 0.0000000451810, 0.000000000000
```

detectors.m

```
loadcmf;  
m = 1000;  
r = zeros(1,m);  
g = zeros(1,m);  
b = zeros(1,m);  
for i=1:length(cmf)  
    r(cmf(i,1)) = cmf(i,2);  
    g(cmf(i,1)) = cmf(i,3);  
    b(cmf(i,1)) = cmf(i,4);  
end  
plot(r,'r',g,'g',b,'b');
```

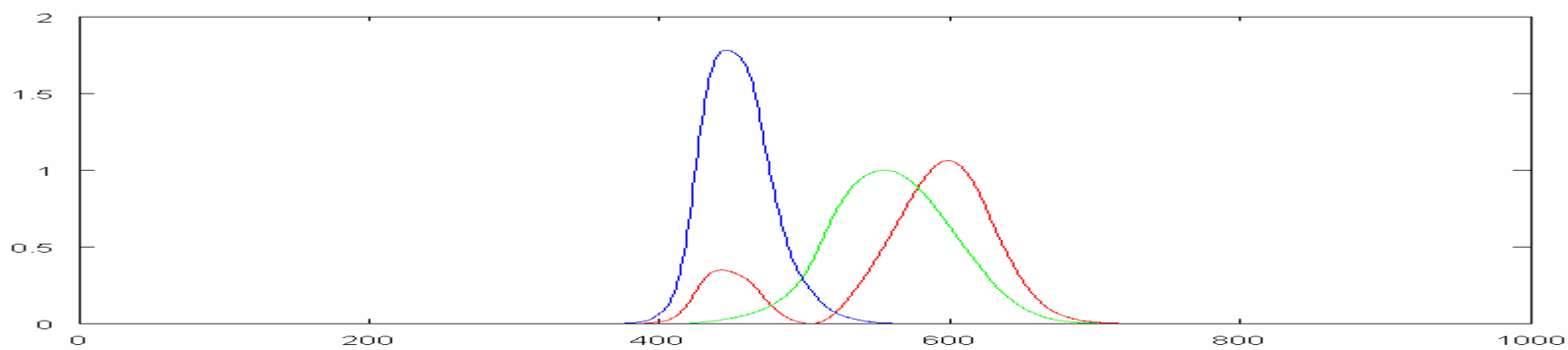
Detektory x,y,z

Citlivosť troch detektorov v našom oku

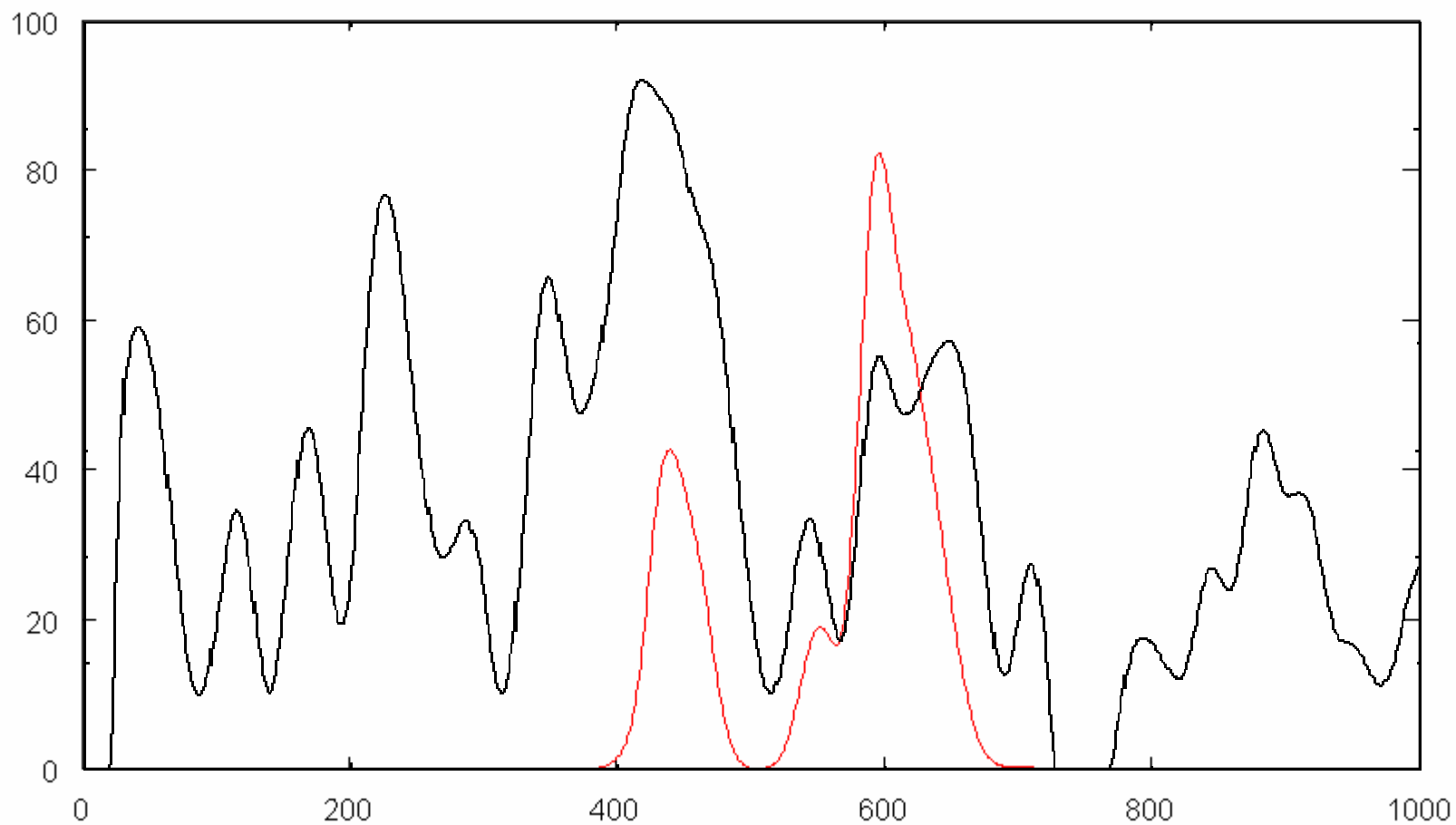


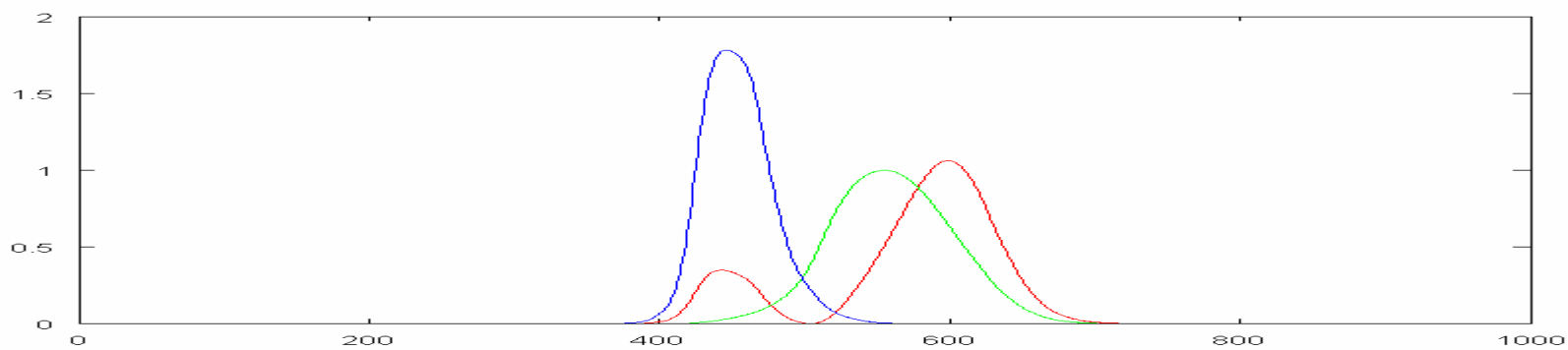
response.m

```
fotons;  
detectors;  
plot(pole.*r,'r');  
pause;  
plot(pole.*g,'g');  
pause;  
plot(pole.*b,'b');  
pause;
```

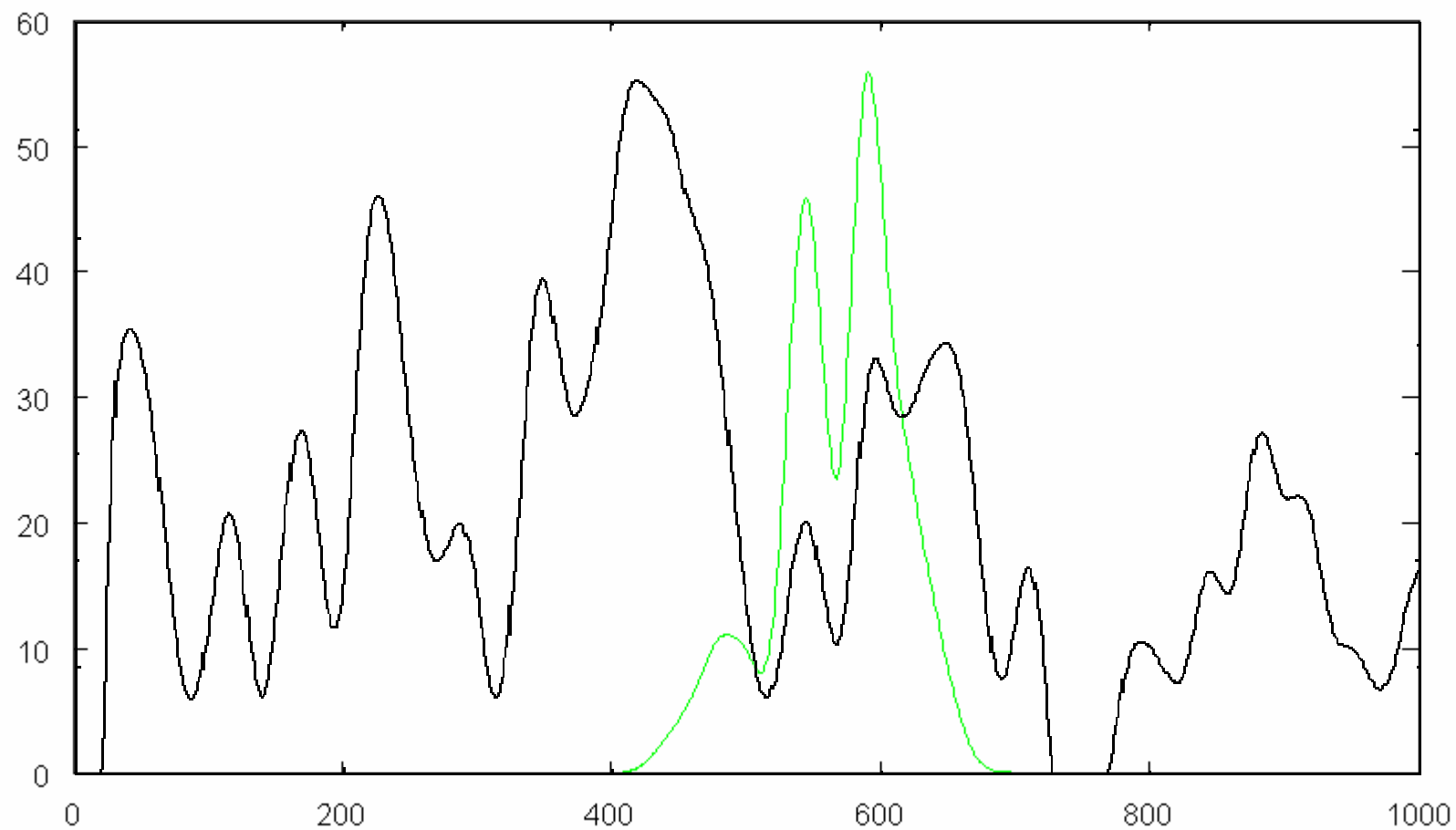


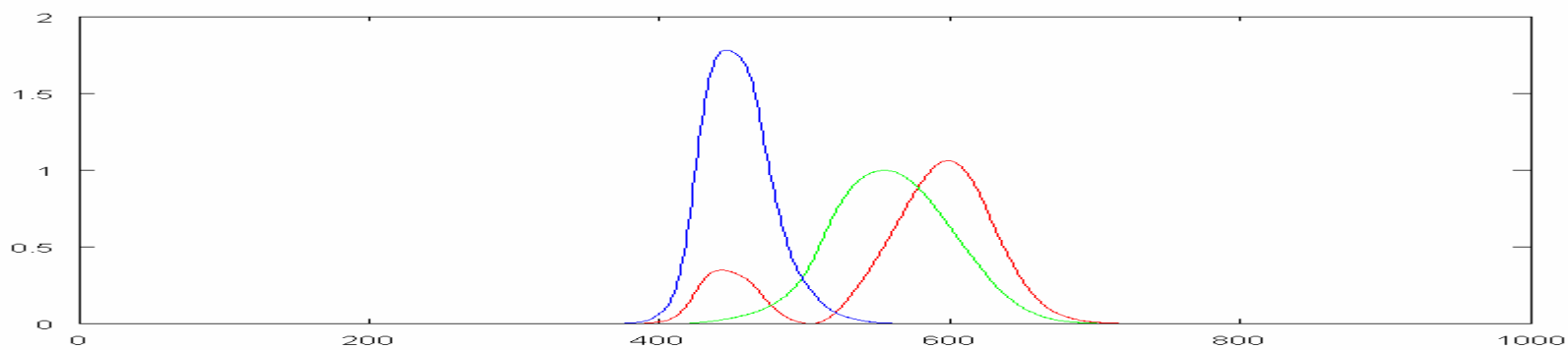
Príklad signálu - odozva prvého detektoru x



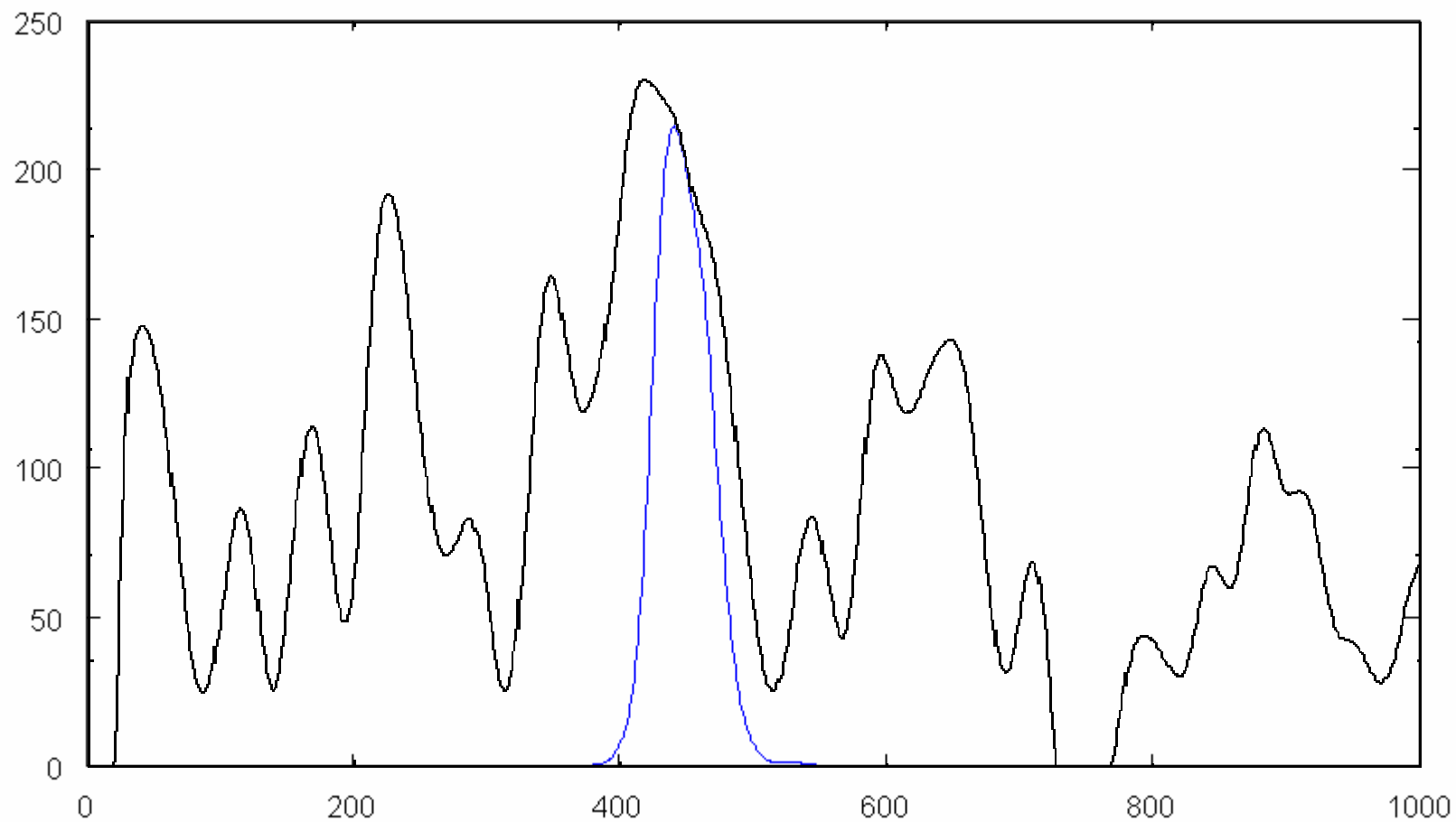


Príklad signálu - odozva druhého detektoru y



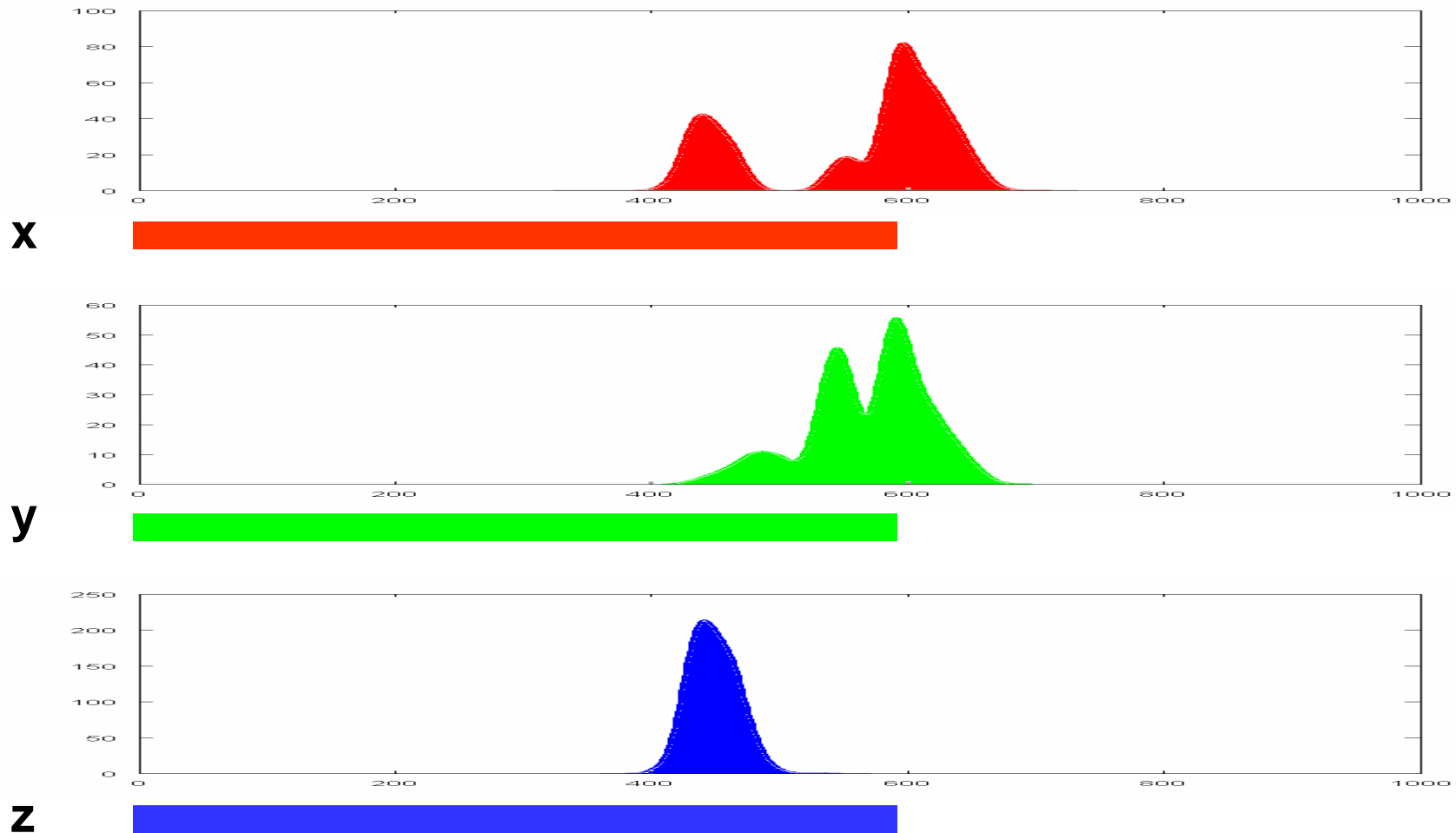


Príklad signálu - odozva tretieho detektoru z



Farba

Výsledkom sú len tri čísla, neberie sa ohľad na to ktoré vlnové dĺžky oko práve dráždia

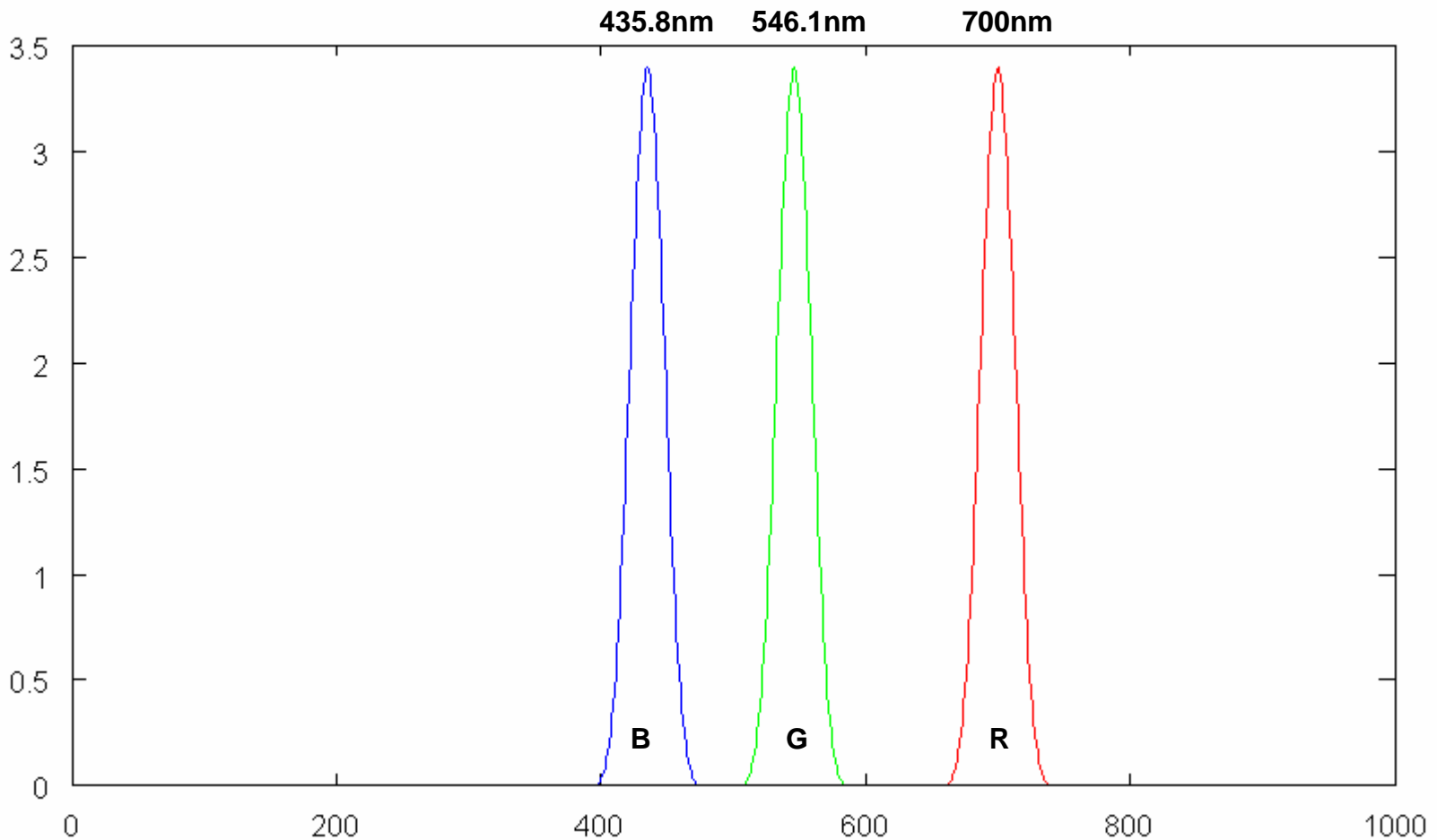


Farba

- Všimnime si, že detektory x, y a z kreslíme červenou, zelenou a modrou farbou
- Pozor, to nie sú detektory červenej, zelenej a modrej farby
- Do mozgu z oka však ide farba v podobe troch čísel

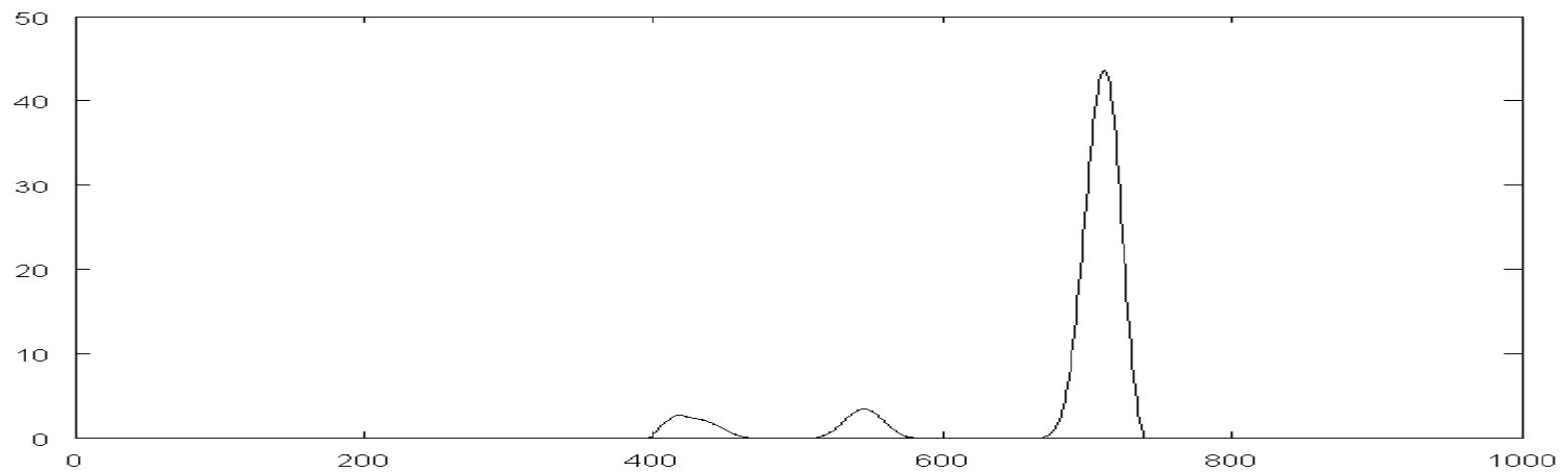
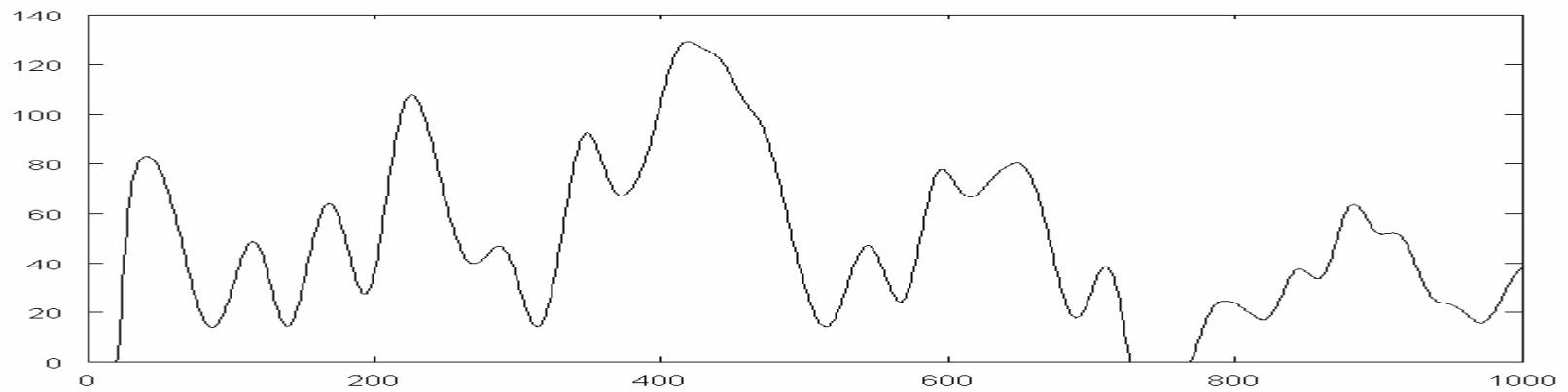
Farba

to využíva monitor počítača, ktorý generuje len tri vlnové dĺžky – červenú, zelenú a modrú – pomocou nich však v mozgu dokáže vyvolať dojem “ľubovoľnej” farby



Farba

Vskutku tieto dve rôzne distribúcie vyvolajú rovnaký vnem



Farba

vnem je rovnaký a je to šedá (aj biela by to mohla byť)

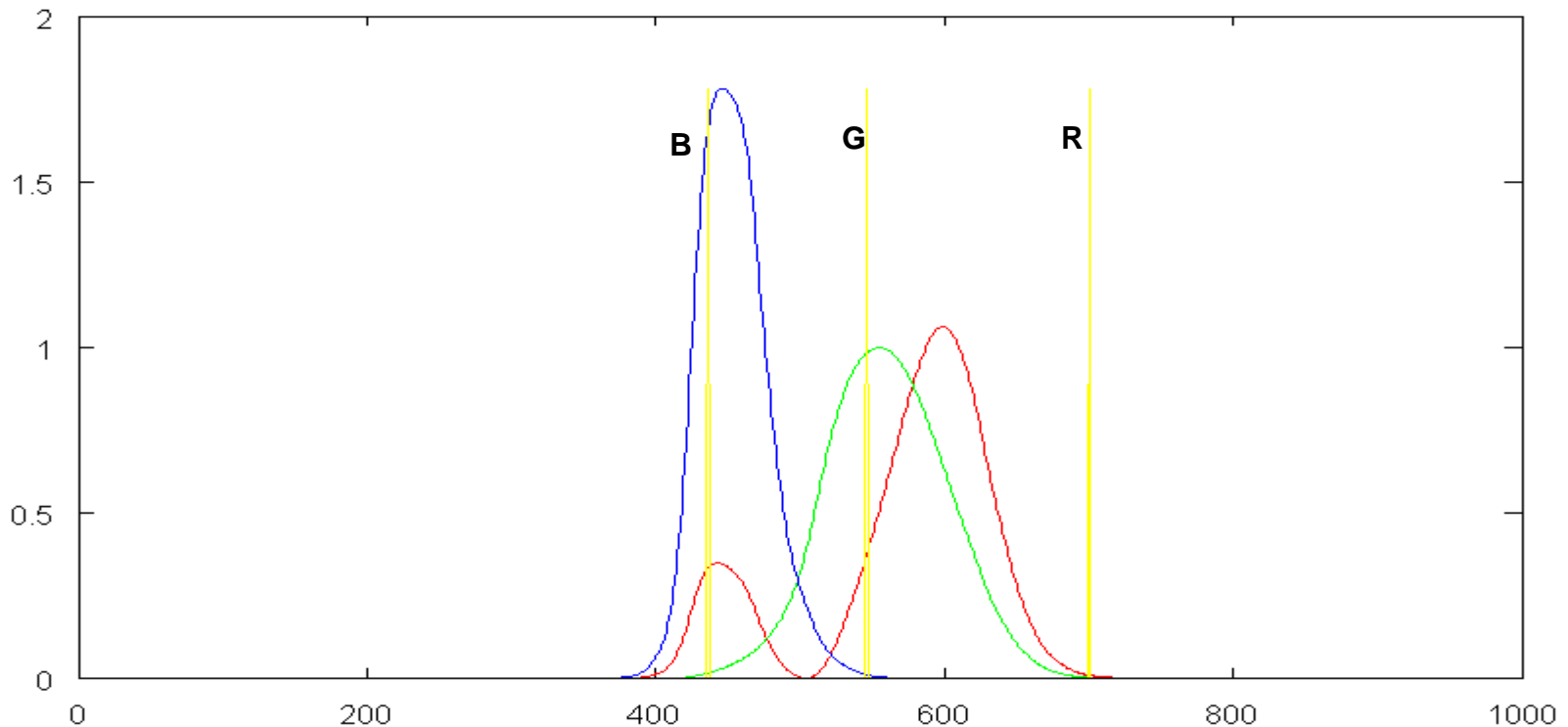


Prečo je to šedá?

- Ako sme vygenerovali signál v našom príklade ?
- Ako priemerovaný náhodný signál z uniformnej distribúcie
- Farba je štatistická vlastnosť distribúcie vlnovej dĺžky fotónov dopadajúcich na bod sietnice

Prečo RGB ?

- Prečo si konštruktéri monitorov vybrali práve červenú 700nm, zelenú 546.1nm a modrú 435.8nm ?



Farebný diagram CIE

- Oko posiela do mozgu vždy tri čísla –
 x, y, z
- Tento priestor môžeme namapovať do
 $\langle 0,1 \rangle \times \langle 0,1 \rangle$ ($X \times Y$)

$$X = x/(x+y+z)$$

$$Y = y/(x+y+z)$$

$$Z = z/(x+y+z) = 1-(X+Y)$$

gamut.m

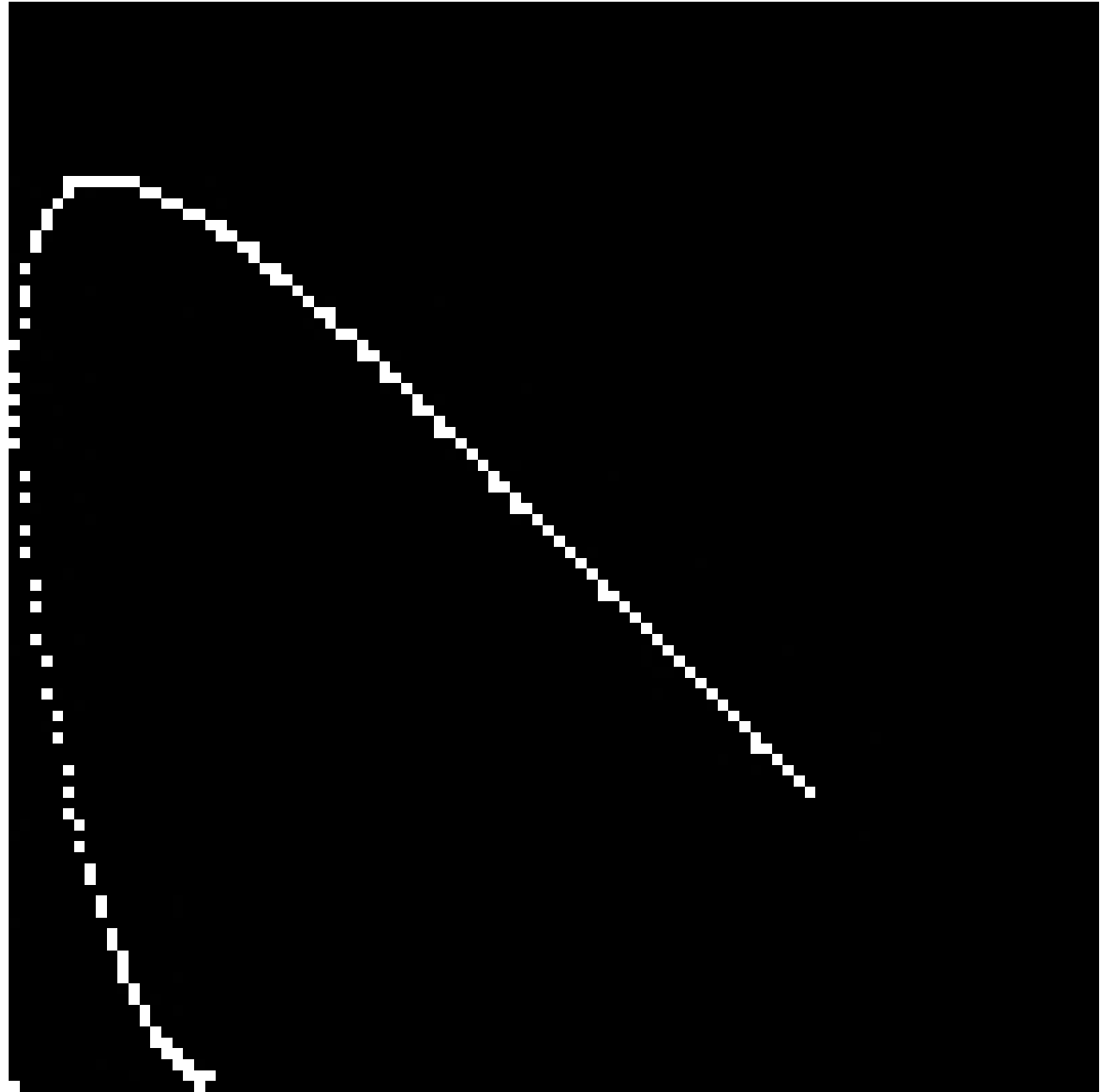
```
loadcmf;
m = 1000;
r = zeros(1,m);
g = zeros(1,m);
b = zeros(1,m);
for i=1:length(cmf)
    r(cmf(i,1)) = cmf(i,2);
    g(cmf(i,1)) = cmf(i,3);
    b(cmf(i,1)) = cmf(i,4);
end
x = zeros(1,m);
y = zeros(1,m);
for i=1:m
    if (r(i)+g(i)+b(i) > 0)
        x(i) = r(i)/(r(i)+g(i)+b(i));
        y(i) = g(i)/(r(i)+g(i)+b(i));
    endif
end

for i=1:m
    x(i) = round(x(i)*100)+1;
    y(i) = round(y(i)*100)+1;
end
plot(x,'r',y,'g');
pause;
obr = zeros(100,100);
for i=1:m
    if (x(i)>0 && y(i)>0)
        obr(101-y(i),x(i)) = 255;
    endif
end
jpgwrite("gamut.jpg",
        obr,obr,obr,100);
```

↑
Y

Monochromatické
svetlo vo
farebnom
diagrame

kde je aká farba?



X →

gamut.m (pokračovanie)

```
mm = max(max(cmf(:,2:4)));
rr = 255*r/mm;
gg = 255*g/mm;
bb = 255*b/mm;
plot(rr,'r',gg,'g',bb,'b');
robr = zeros(100,100);
gobr = zeros(100,100);
boobr = zeros(100,100);
for i=1:m
    if (x(i)>0 && y(i)>0)
        robr(101-y(i),x(i)) = rr(i);
        goobr(101-y(i),x(i)) = gg(i);
        boobr(101-y(i),x(i)) = bb(i);
    endif
end
jpgwrite("gamut2.jpg",roobr,gobr,boobr,100);
```

↑
Y

Monochromatické
svetlo vo
farebnom
diagrame podľa
farby

kde je biela?
kde je červená?
kde je modrá?
kde je zelená?



X →

gamut.m (pokračovanie)

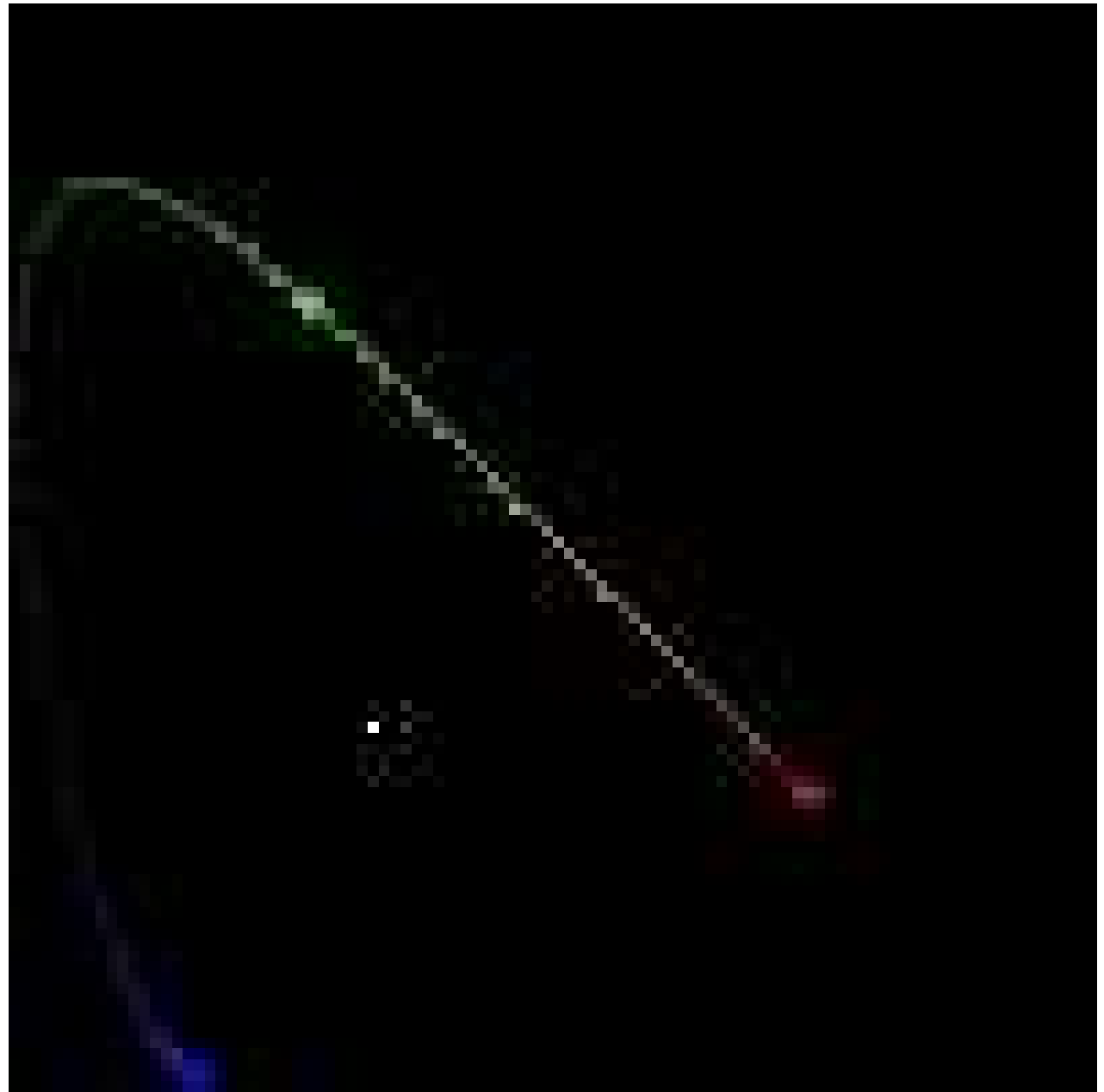
```
xx = sum(r)/(sum(r)+sum(g)+sum(b));
yy = sum(g)/(sum(r)+sum(g)+sum(b));
xx = round(100*xx)+1;
yy = round(100*yy)+1;
robr(101-yy,xx) = 255;
gobr(101-yy,xx) = 255;
bobr(101-yy,xx) = 255;

r1=700;
xx = r(r1)/(r(r1)+g(r1)+b(r1));
yy = g(r1)/(r(r1)+g(r1)+b(r1));
xx = round(100*xx)+1;
yy = round(100*yy)+1;
robr(101-yy,xx) = 255;
gobr(101-yy,xx) = 0;
bobr(101-yy,xx) = 0;
% aj pre g1=546 a b1=436
jpgwrite("gamut3.jpg",robr,gobr,bobr,100);
```

Reálne monitor
nevysiela čisto
monochromatické
svetlo, preto
skutočné RGB
padne dovnútra,
bližšie k bielej

za možnosť
vyjadriť farby v
lineárnom
priestore
zaplatíme ďalším
obmedzením
farieb.

↑
Y



X →

Vyjadríme s RGB ľubovoľnú farbu ?

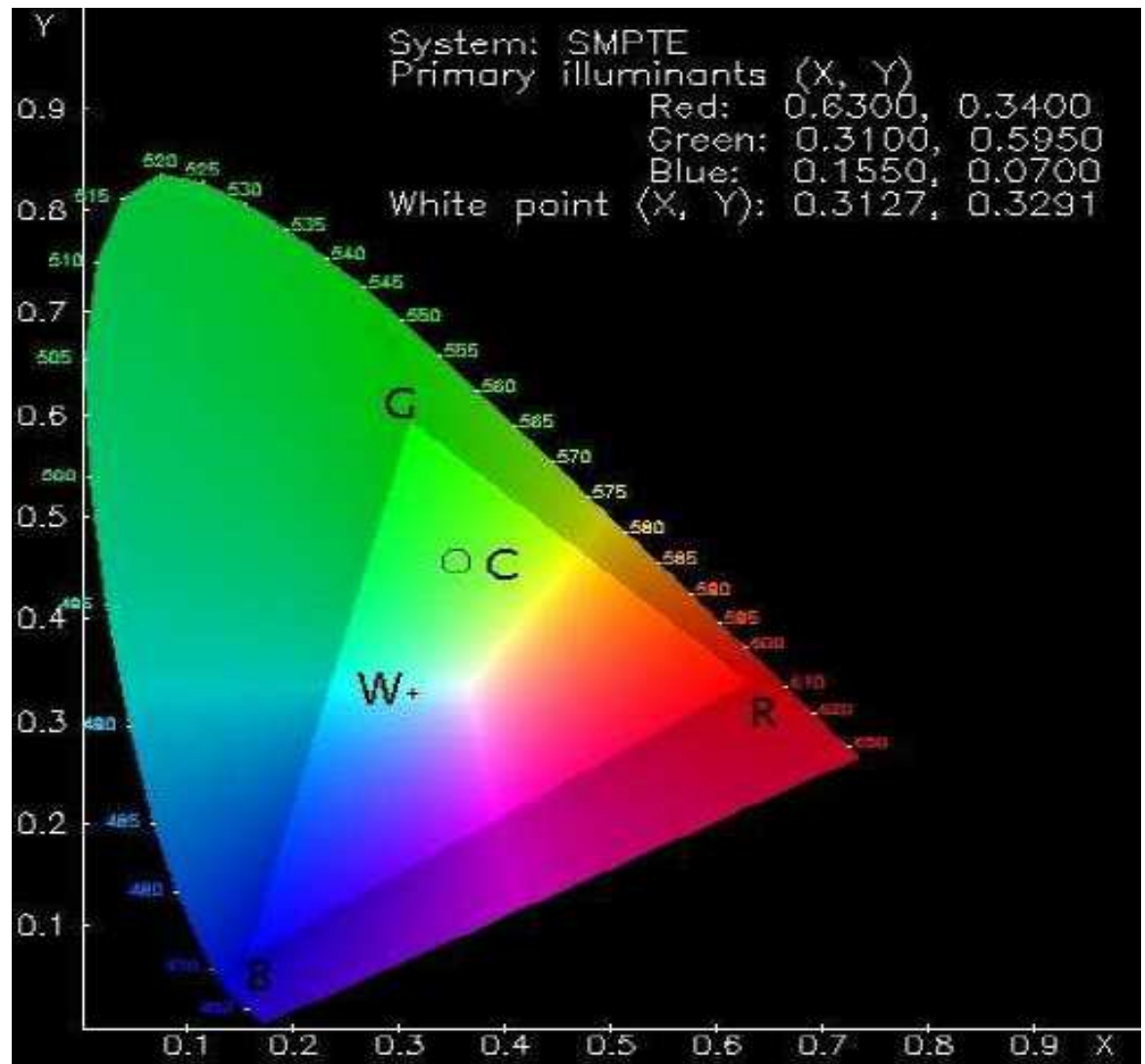
Nie celkom

Navyše trochu
iné RGB má
NTSC a iné
napr. PAL

$$X = 2.77R + 1.75G + 1.13B$$

$$Y = 1.00R + 4.59G + 0.06B$$

$$Z = 0.00R + 0.06G + 5.59B$$



Farebné modely

Monitor potrebuje tzv. aditívny farebný model, lebo je zdrojom svetla

Iný model potrebujeme pri farebnej tlači, kde sa od bieleho papiera svetlo odráža a my tam tlačíme filtre ktoré zabráňujú jeho príchodu do našich očí. Vtedy sa hodí fialová, tyrkízová a žltá. (subtraktívny model)

Dúha

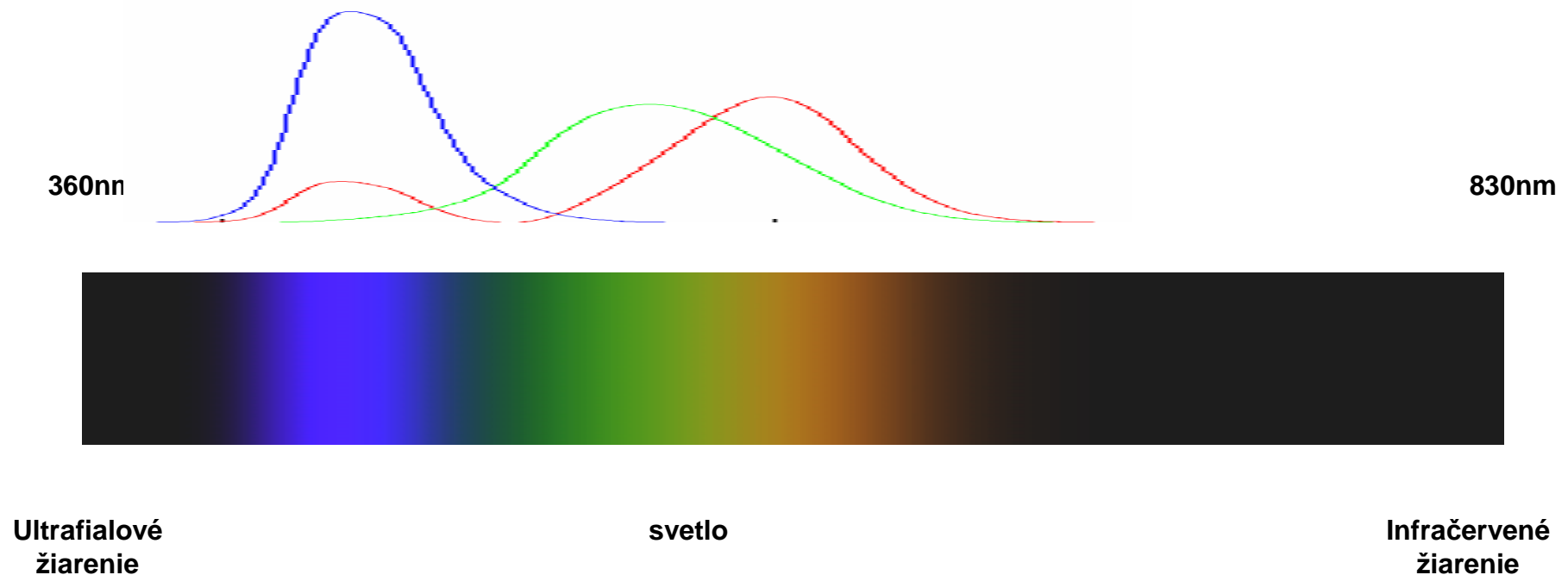
- vzniká rozkladom svetla ma monochromatické zložky.
- z dát CIE pre jednotlivé detektory ju preto možno približne spočítať v RGB

rainbow.m

```
loadcmf;  
m = max(max(cmf(:,2:4)));  
r = 255*repmat(cmf(:,2),1,100)/m;  
g = 255*repmat(cmf(:,3),1,100)/m;  
b = 255*repmat(cmf(:,4),1,100)/m;  
jpgwrite("duha.jpg",r,g,b,100);
```

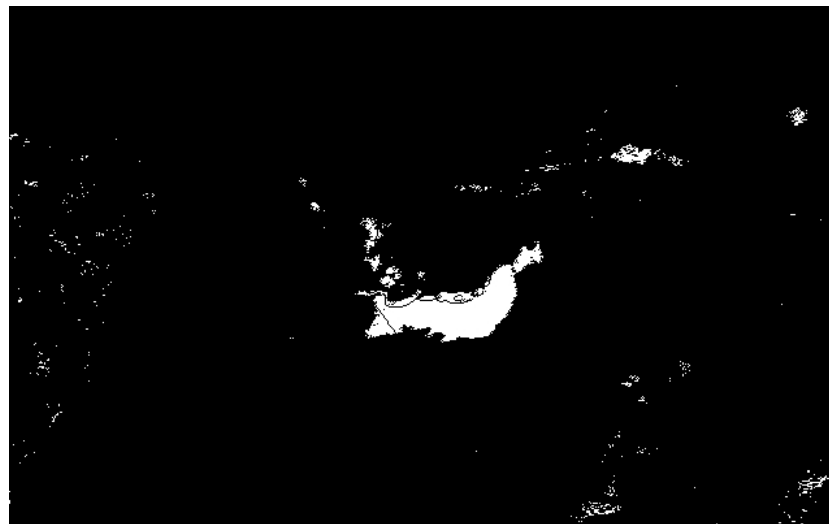
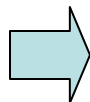
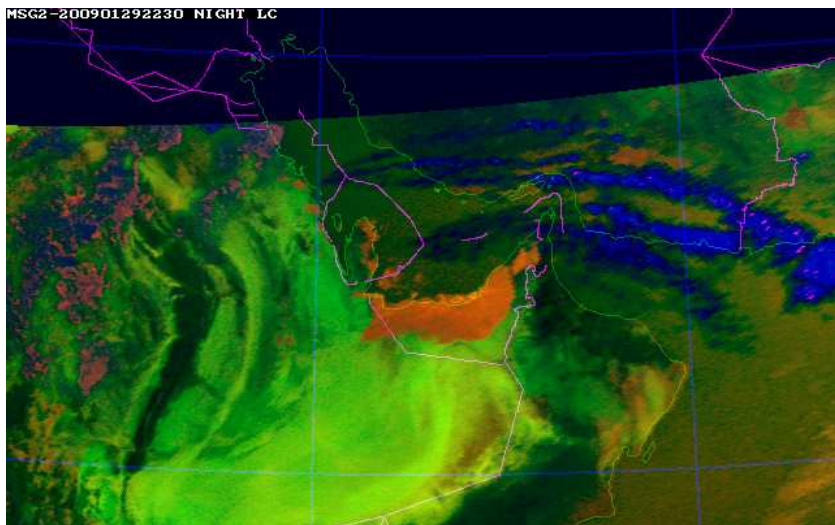
Dúha

- Z citlivosti detektorov ľahko odvodíme ako bude vyzerat' dúha - spektrum mono-chromatického svetla



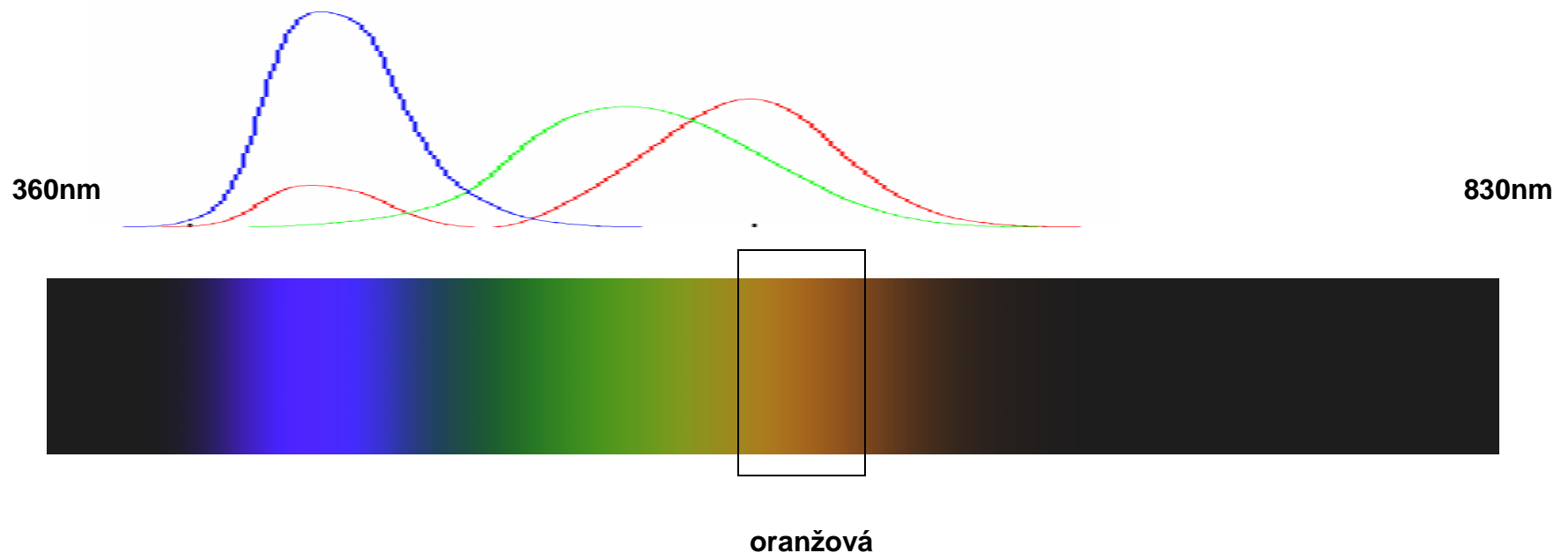
Príklad rozpoznania farby

- Ako rozpoznať oranžovú farbu?



Dúha

- Na dúhe je vidno oranžovú farbu a vidíme:
 - neobsahuje modrú zložku
 - obsahuje proporcionálne viac červenej ako zelenej



(Pozor: nie každá farba je na dúhe, napr. šedá)

Model

- if (blue $< c_1$ && $c_2 < \text{red/green} < c_3$)
then orange
- kde c_1 je malé a $c_2 < 2 < c_3$
- Aké sú ale c_1, c_2, c_3 ?
- Z fyzikálneho modelu to nezistíme
- Dôvod: nik nevie kde je hranica medzi
žltou, oranžovou a červenou
- Musíme pozrieť normu – dohodu o
hraniciach oranžovej

Norma

[http://en.wikipedia.org/wiki/Orange_\(colour\)](http://en.wikipedia.org/wiki/Orange_(colour))

http://en.wikipedia.org/wiki/Variations_of_orange

Orange (web colour)
— Colour coordinates —

Hex triplet #FFA500

sRGB^B (r, g, b) (255, 165, 0)

HSV (h, s, v) (39°, 100%, 100%)

Source [CSSX11/SVG^{\[2\]}](#)

B: Normalized to [0–255] (byte)

Tangerine
— Colour coordinates —

Hex triplet #F28500

RGB^B (r, g, b) (243, 133, 0)

HSV (h, s, v) (33°, 100%, 95%)

Source [\[Un sourced\]](#)

B: Normalized to [0–255] (byte)

Brown
— Colour coordinates —

Hex triplet #964B00

RGB^B (r, g, b) (150, 75, 0)

HSV (h, s, v) (30°, 100%, 59%)

Source [\[Un sourced\]](#)

B: Normalized to [0–255] (byte)

Dark orange (web colour)
— Colour coordinates —

Hex triplet #FF8C00

sRGB^B (r, g, b) (255, 140, 0)

HSV (h, s, v) (34°, 100%, 100%)

Source [X11/SVG^{\[2\]}](#)

B: Normalized to [0–255] (byte)

Carrot orange
— Colour coordinates —

Hex triplet #ED9121

RGB^B (r, g, b) (237, 145, 33)

HSV (h, s, v) (33°, 86%, 93%)

Source [\[Un sourced\]](#)

B: Normalized to [0–255] (byte)

...

Orange peel
— Colour coordinates —

Hex triplet #FFA000

RGB^B (r, g, b) (255, 160, 0)

HSV (h, s, v) (38°, 100%, 100%)

Source [\[Un sourced\]](#)

B: Normalized to [0–255] (byte)

Burnt orange
— Colour coordinates —

Hex triplet #CC5500

RGB^B (r, g, b) (204, 85, 0)

HSV (h, s, v) (25°, 100%, 80%)

Source [University of Texas at Austin^{\[7\]}](#)

B: Normalized to [0–255] (byte)

```
orange = [  
255, 229, 180;  
251, 206, 177;  
255, 153, 102;  
254, 111, 94;  
248, 131, 121;  
255, 160, 0;  
255, 165, 0;  
255, 127, 0;  
255, 140, 0;  
243, 132, 0;  
255, 117, 24;  
249, 77, 0;  
237, 145, 33;  
255, 150, 0;  
204, 85, 0;  
150, 75, 0  
];
```

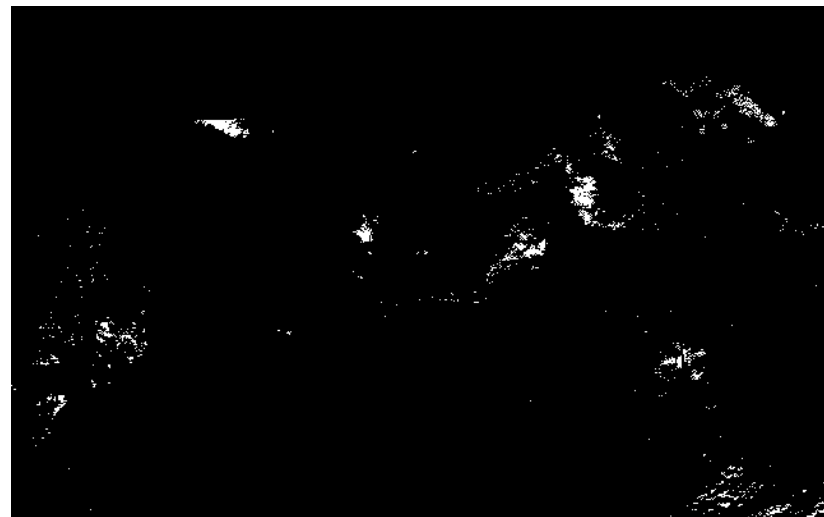
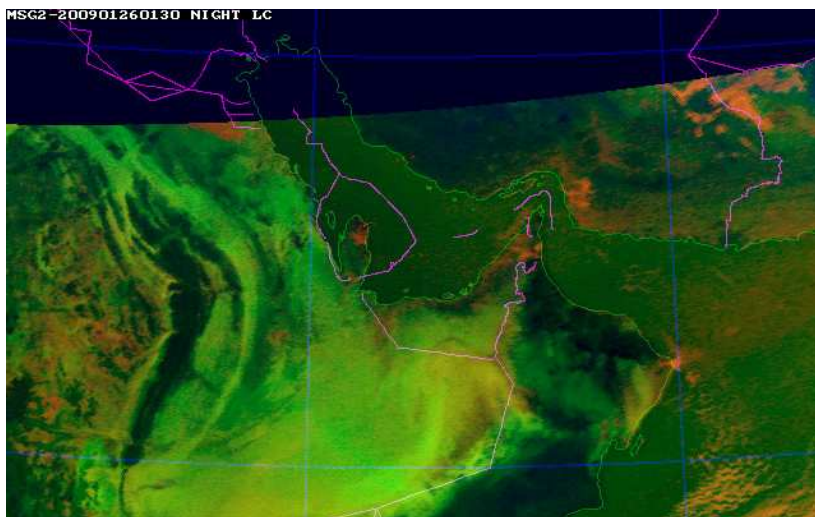
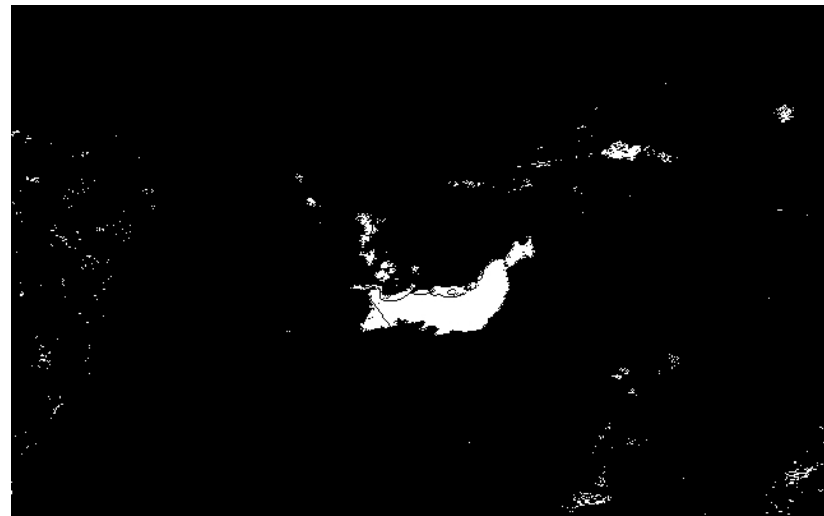
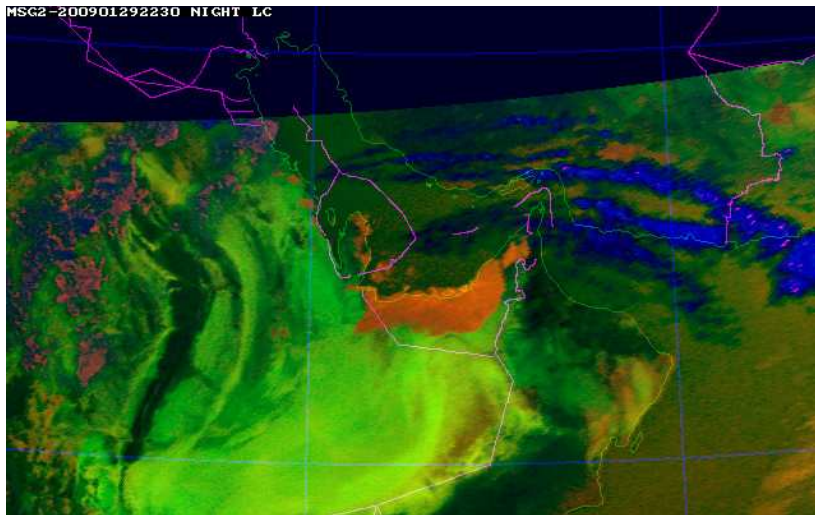
Metóda

- Podľa normy vypočítame parametre modelu:
- if (blue < 35 && 1.1 < red/green < 3.3)
then orange
- A musíme ešte zabrániť aby sa príliš čierne bárs pomermi oranžové bralo v úvahu: red > 50
- (red, green, blue ∈ <0,255>)

fog.m

```
fil="MSG2-d02-434-200901292230.jpg";
img = imread(fil);
sz = size(img);
for i=1:sz(1)
    for j=1:sz(2)
        r = img(i,j,1);
        g = img(i,j,2);
        b = img(i,j,3);
        c = r/(g+1);
        if (r > 50 && c > 1.1 && c < 3.3 && b < 35)
            img(i,j,:) = [255,255,255];
        else
            img(i,j,:) = [0,0,0];
        endif
    end
end
jpgwrite([ "org-" fil ],img(:,:,1),img(:,:,2),img(:,:,3),100);
```

Výsledky



Ďakujem za pozornosť!

Andrej Lúčný

www.microstep-mis.com/~andy