

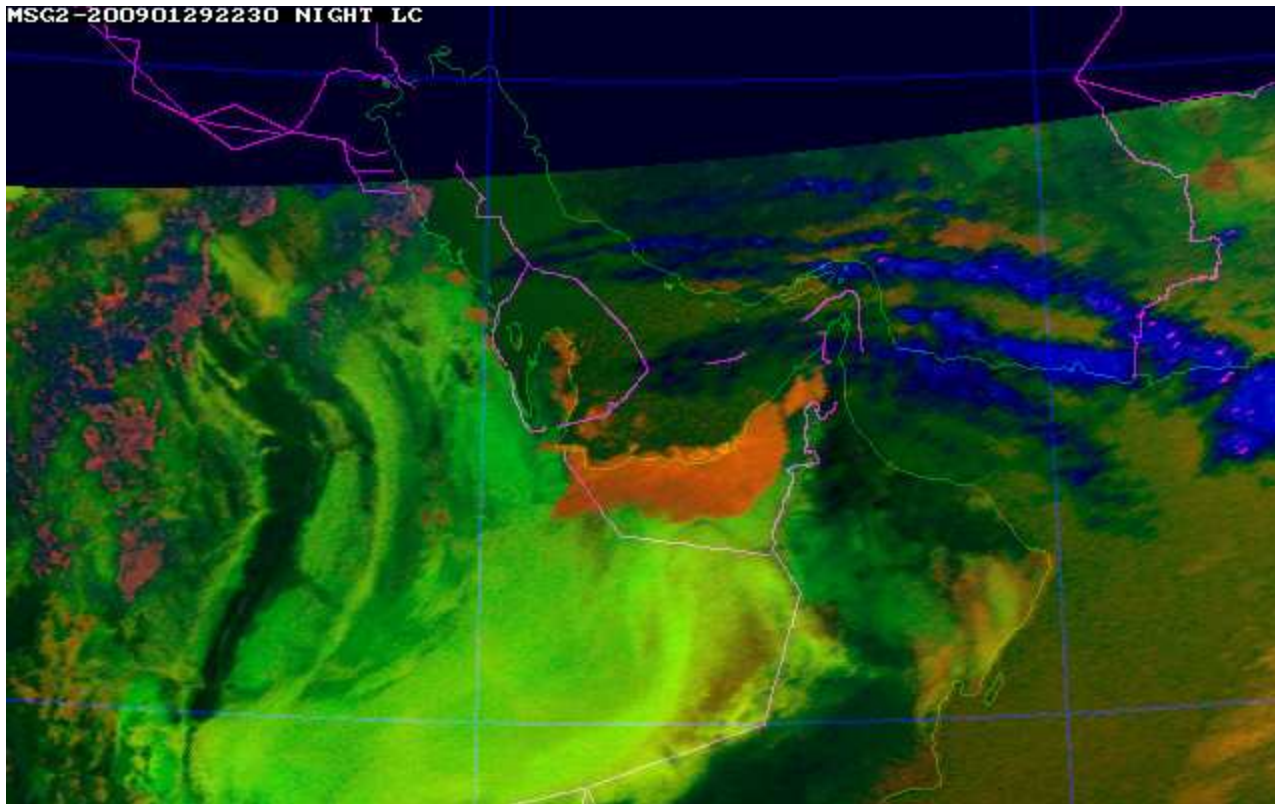
# Ako môže počítač poznať oranžovú farbu ?

Robotický seminár  
robotika.sk

Andrej Lúčny  
[www.microstep-mis.com/~andy](http://www.microstep-mis.com/~andy)

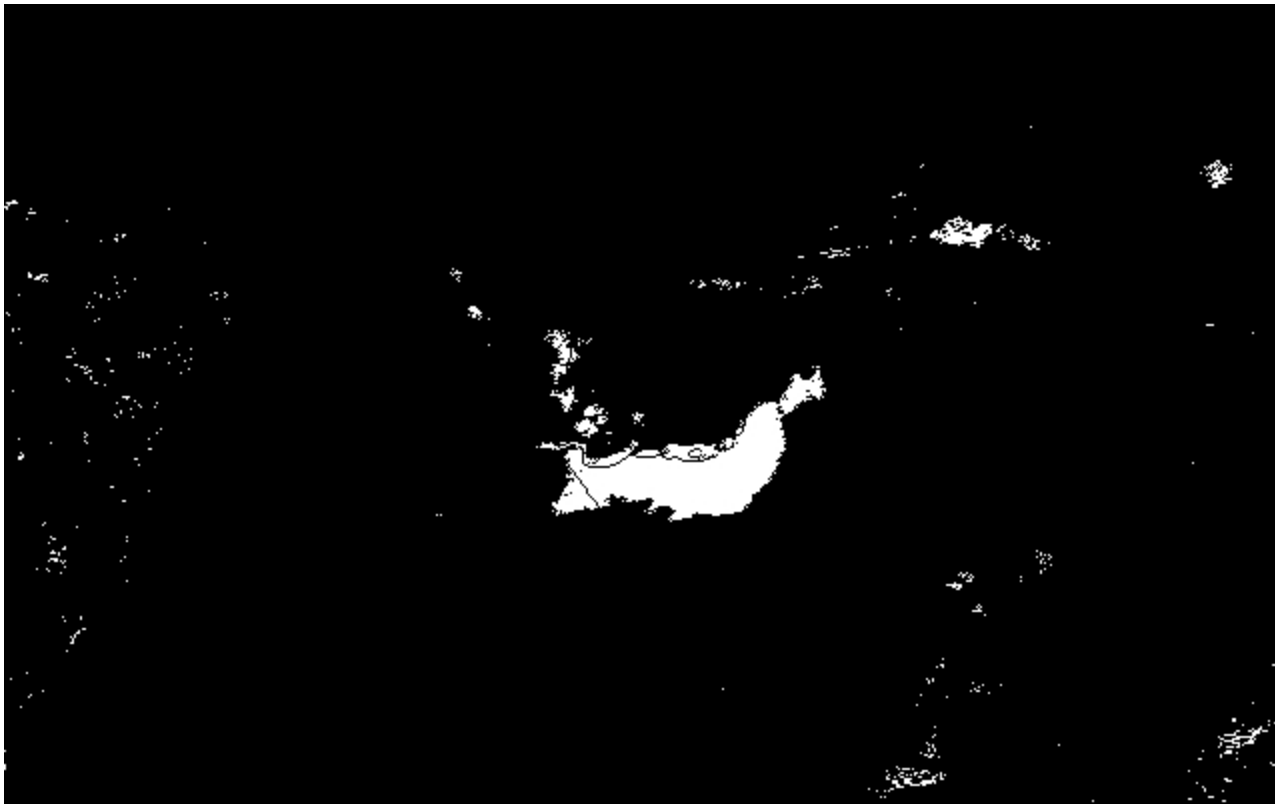
# Úloha

- Z družicového snímku treba extrahovať hmlu označenú rôznymi odtieňmi oranžovej farby



# Úloha

- cieľom je analogický obrázok 0/1 kde 1 je hmla



Ako naučiť počítač čo je oranžová farba ?

- Možností je viacero
- Nápady ?

Ako naučiť počítač čo je oranžová farba ?

- označiť hmlu ručne a snažiť sa využiť adaptívne vycvičiť nejaký klasifikátor
- Nejde to. Príšerne pracné, nedokážeme sa rozhodnúť čo je ešte oranžové a čo už nie

Ako naučiť počítač čo je oranžová farba ?

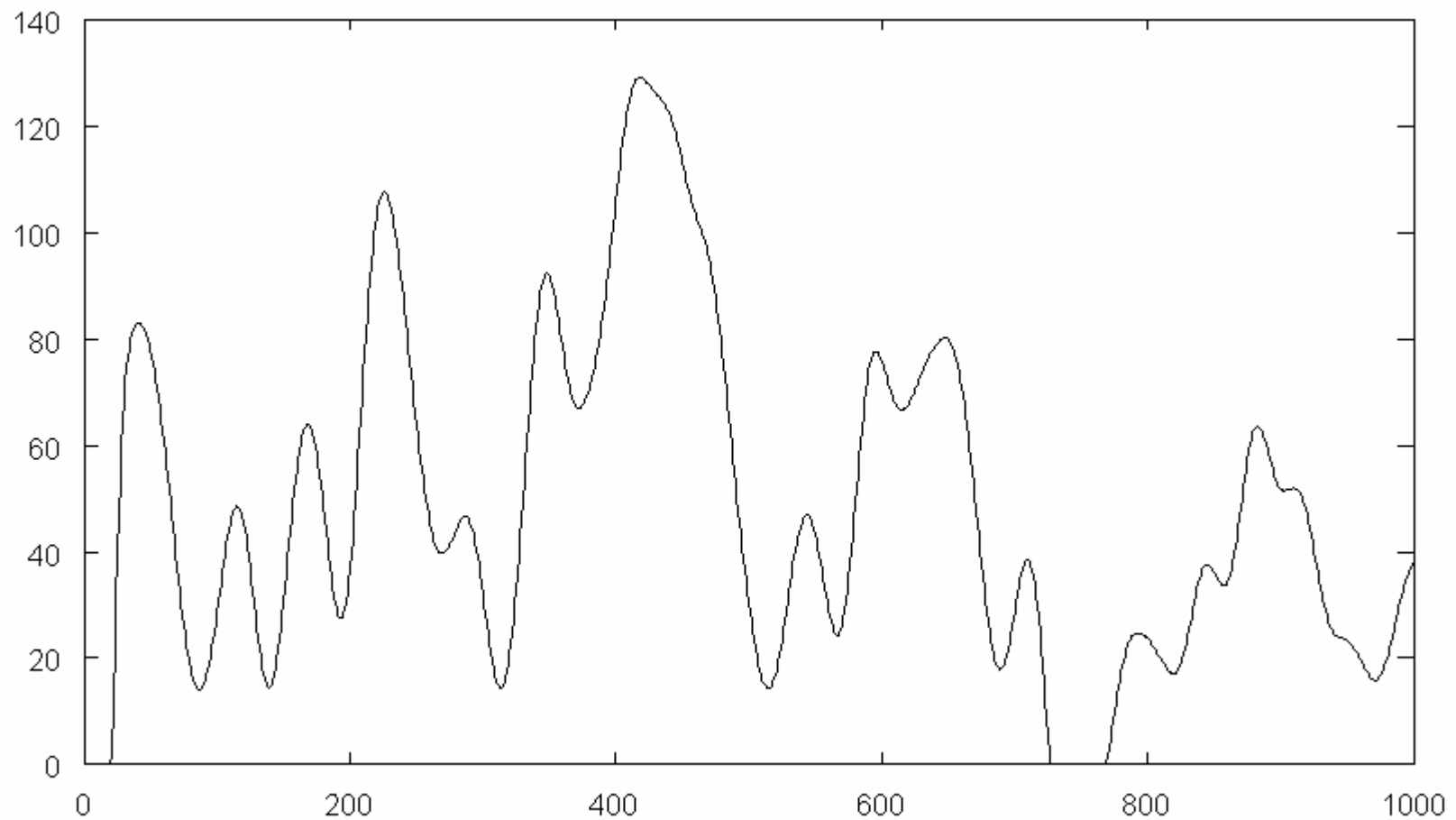
- Fyzika:
- čo sú farby?
- čo je oranžová?
  
- Povedzme si to, ale vopred prezradíme, že samo o sebe to na riešenie nestačí. Konečné riešenie však bude fyzikálny model využívať

# Farby

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

# Farby

Príklad signálu - počty fotónov (v zmyslenej jednotke)





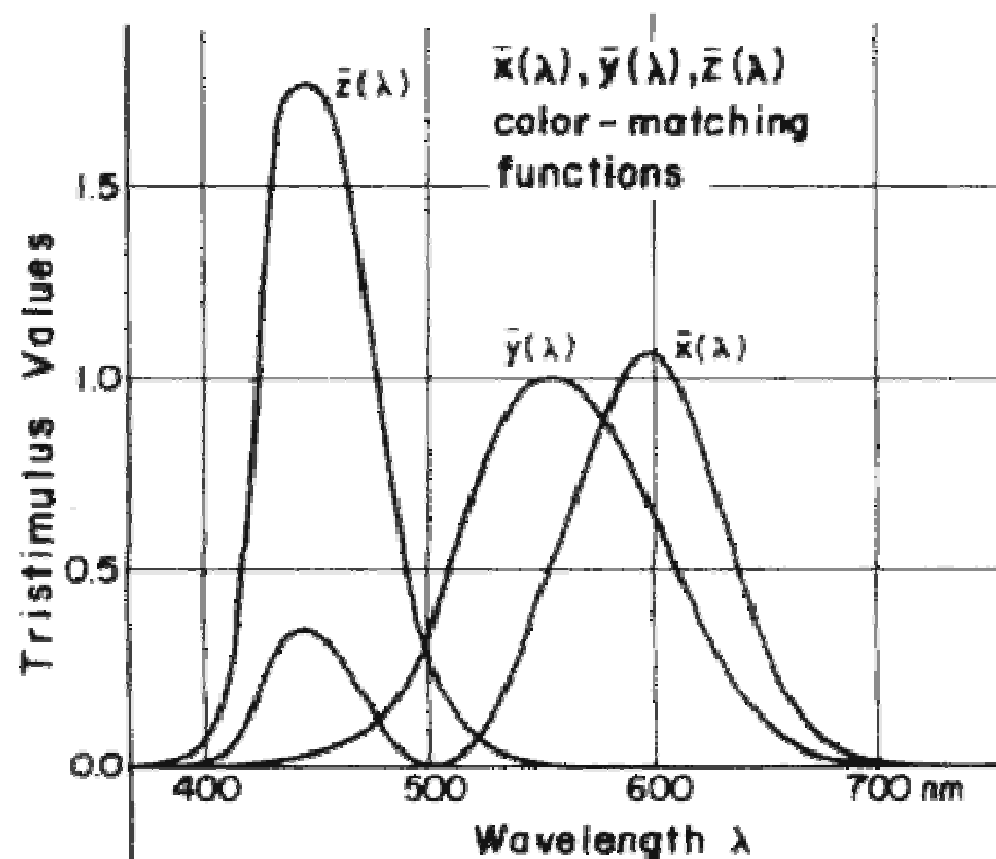
# Farby

- Na sietnici oka sú v jednom „bode“ bunky 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

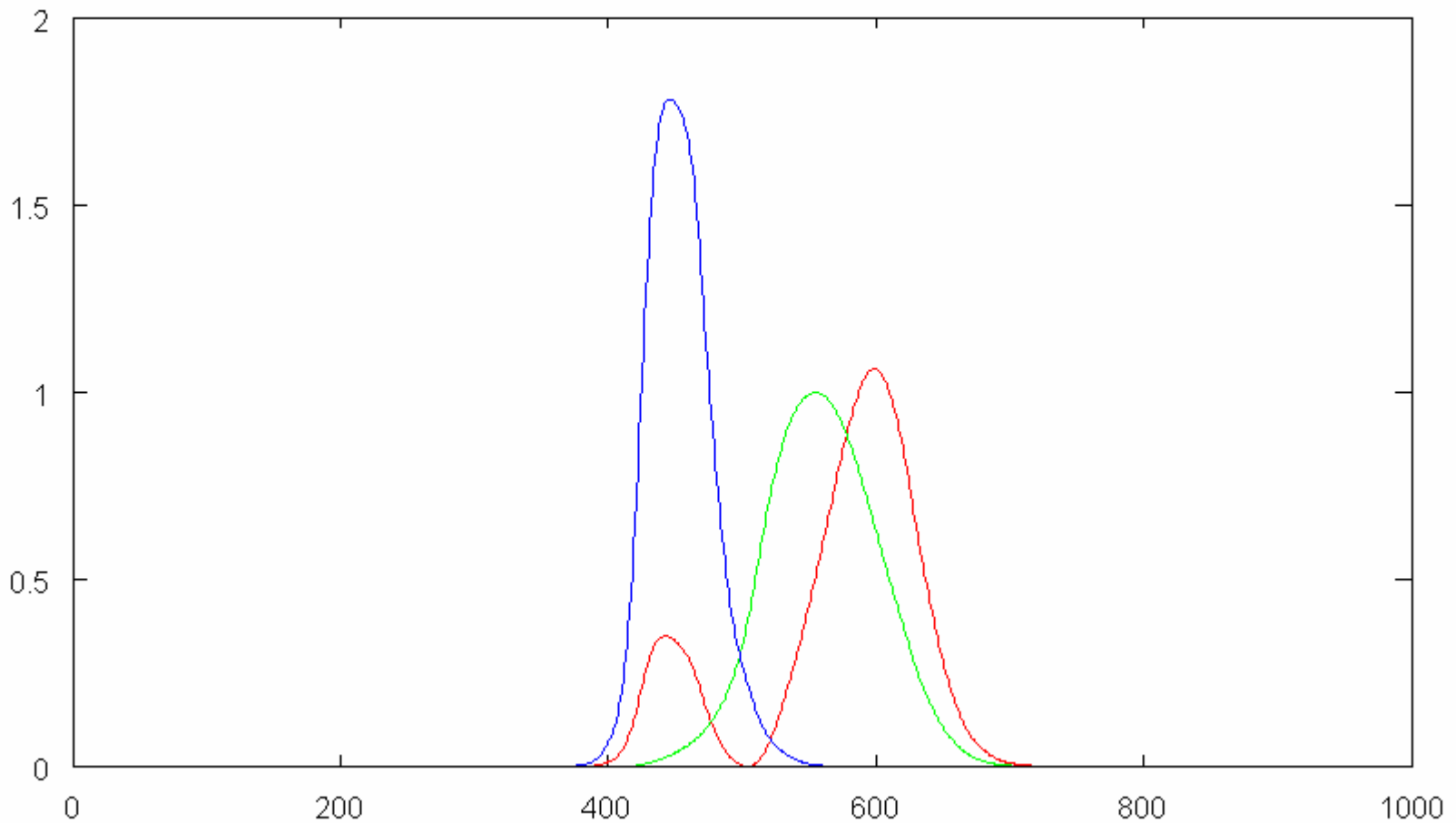
(X, Y, Z) - System

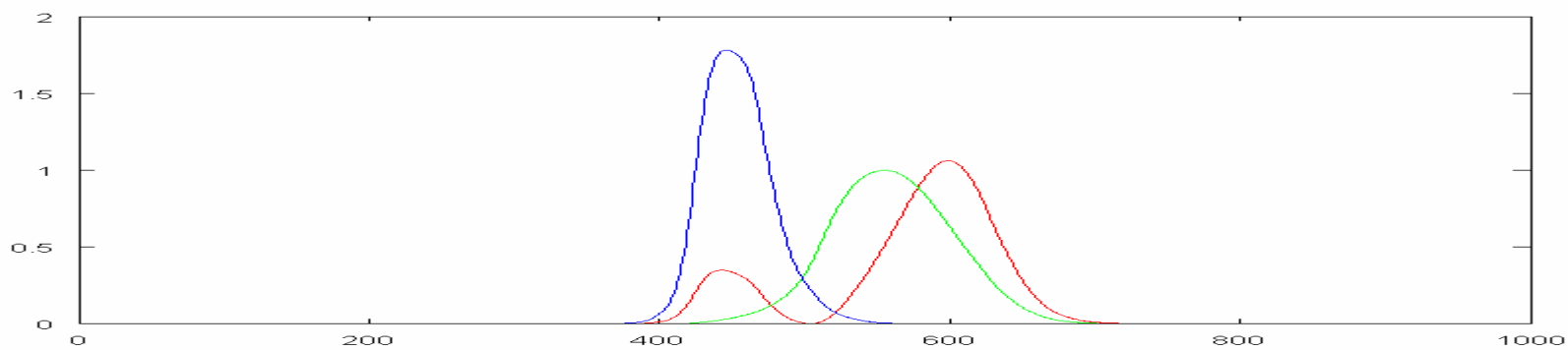


CIE 1931

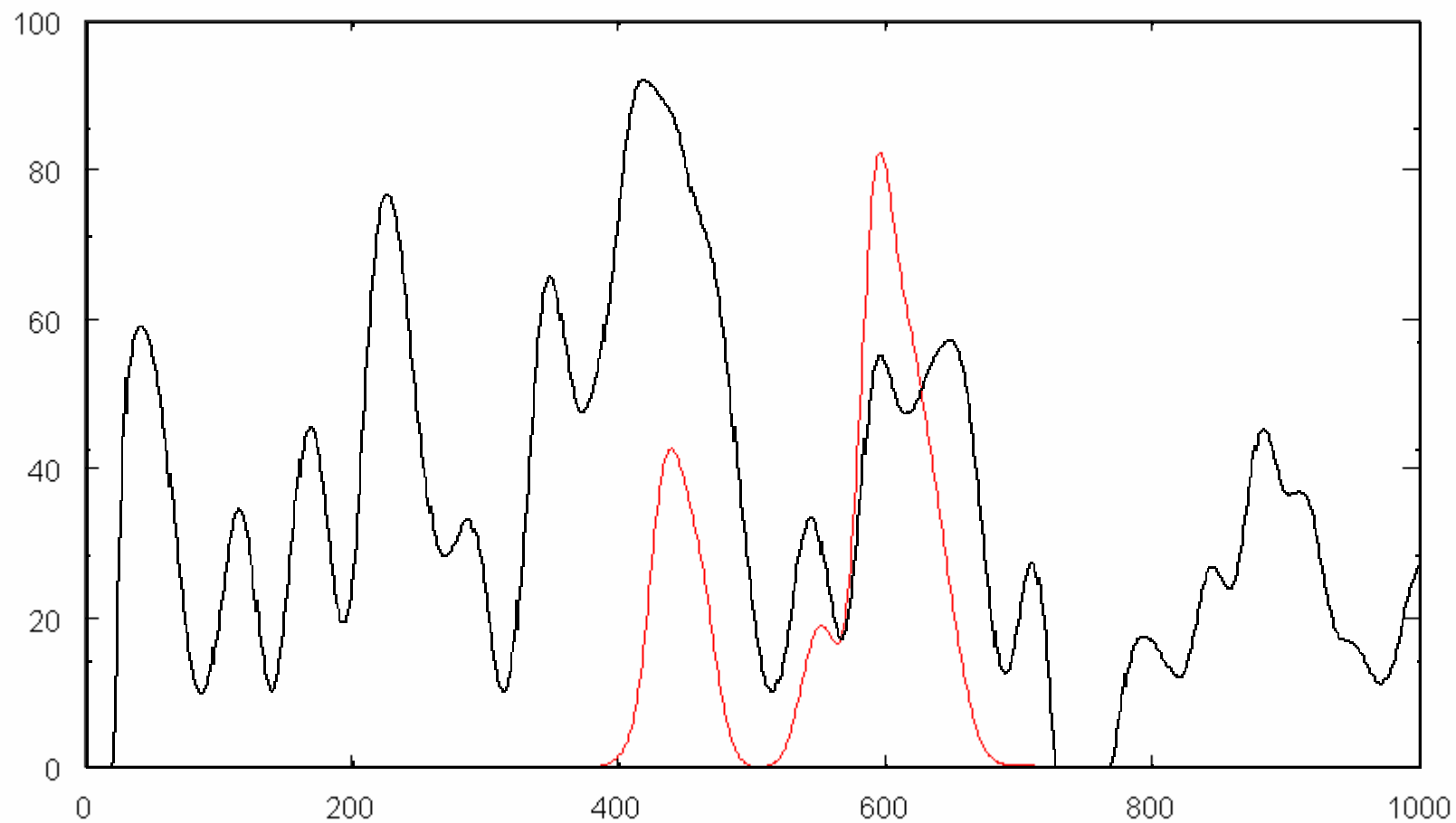
# Farby

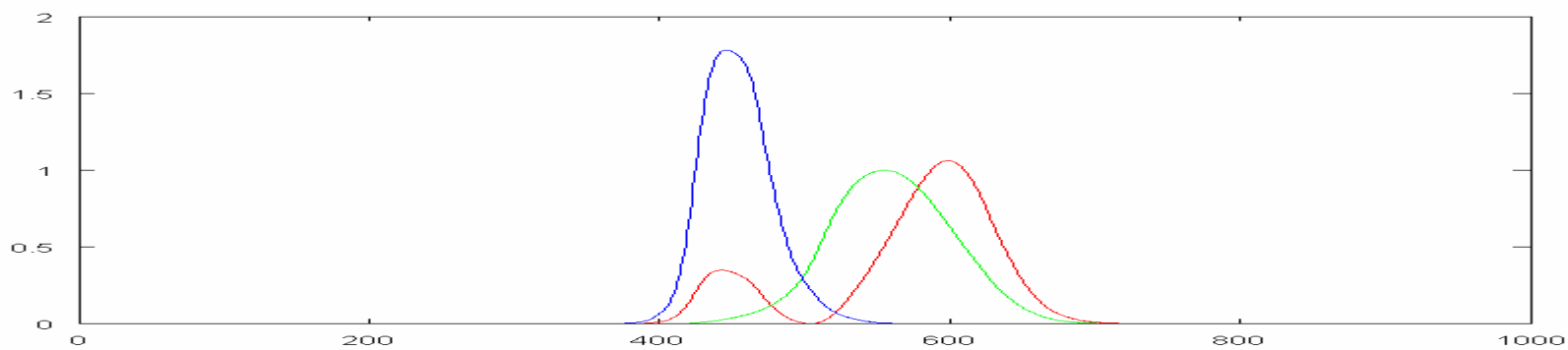
Citlivosť troch detektorov v našom oku



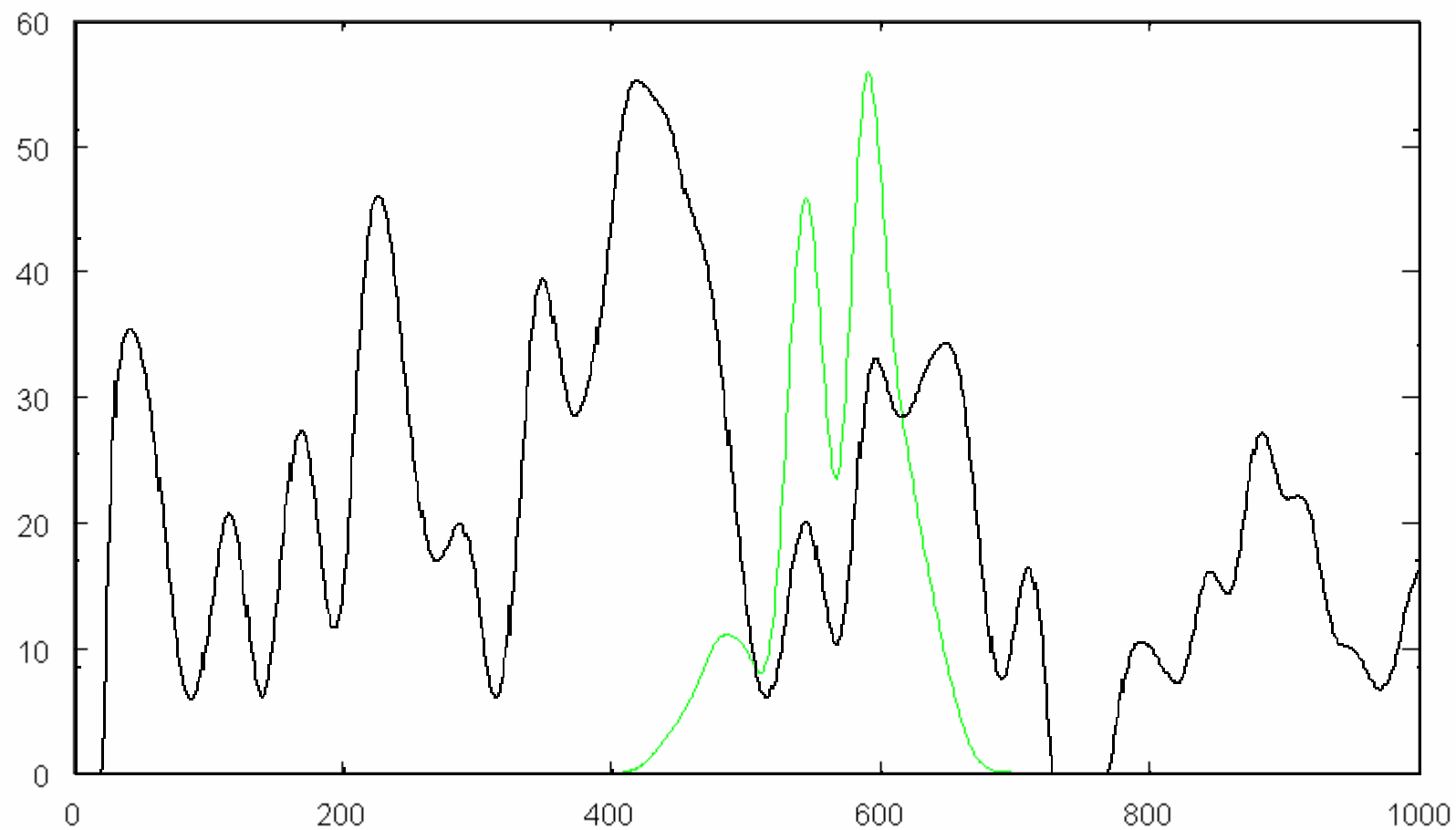


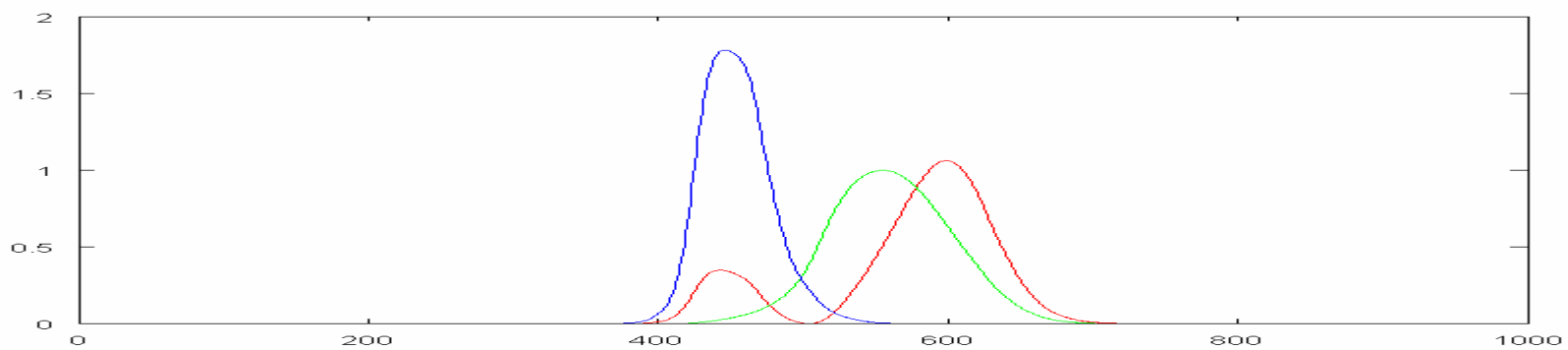
## Príklad signálu - odozva prvého detektoru



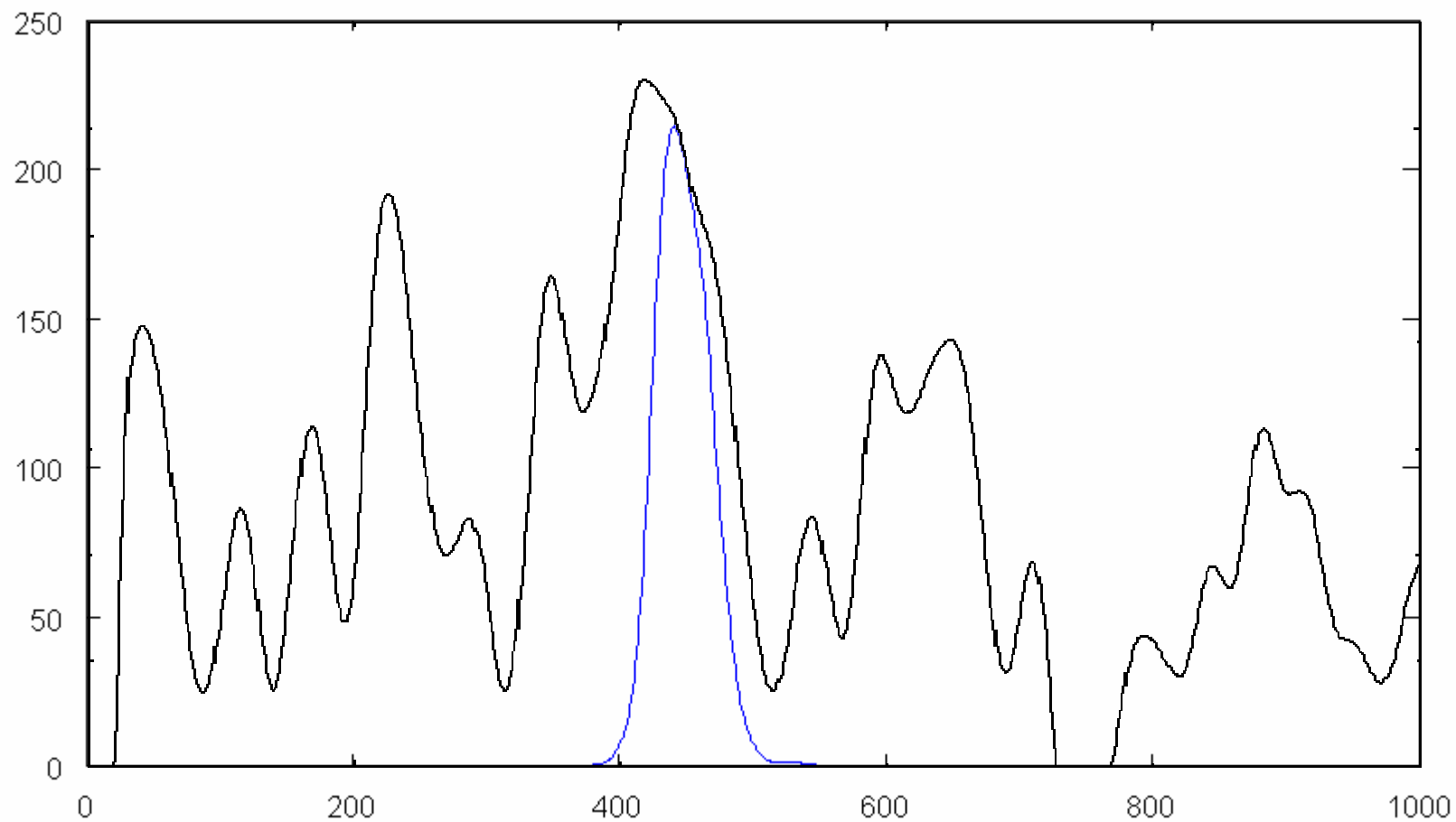


## Príklad signálu - odozva druhého detektoru



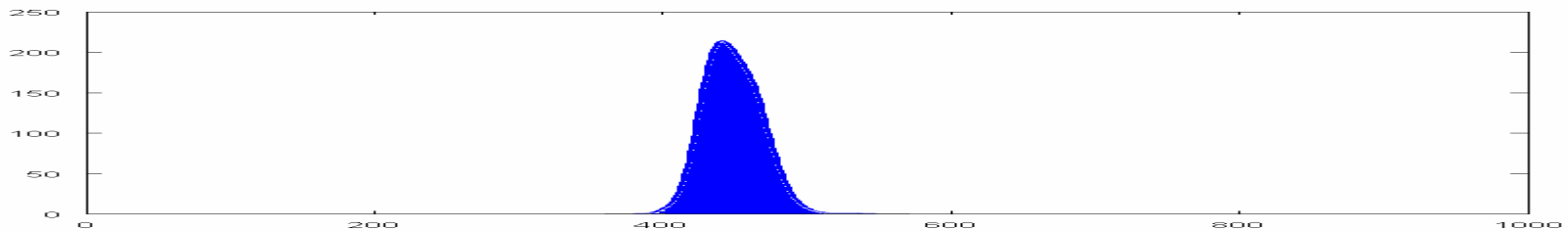
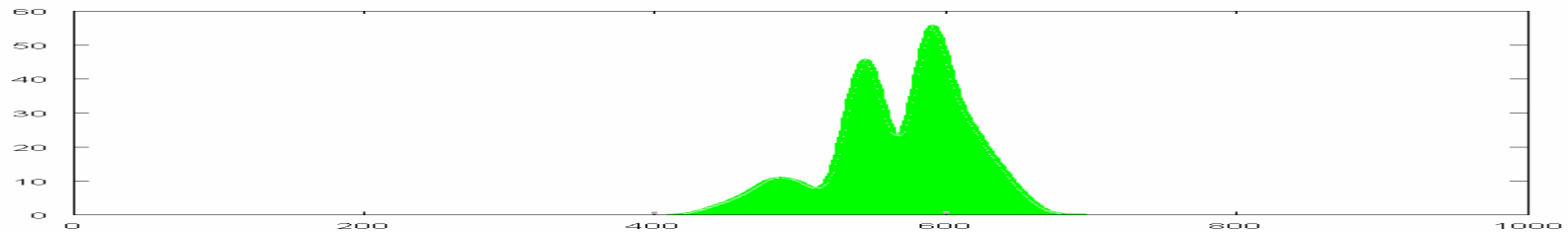
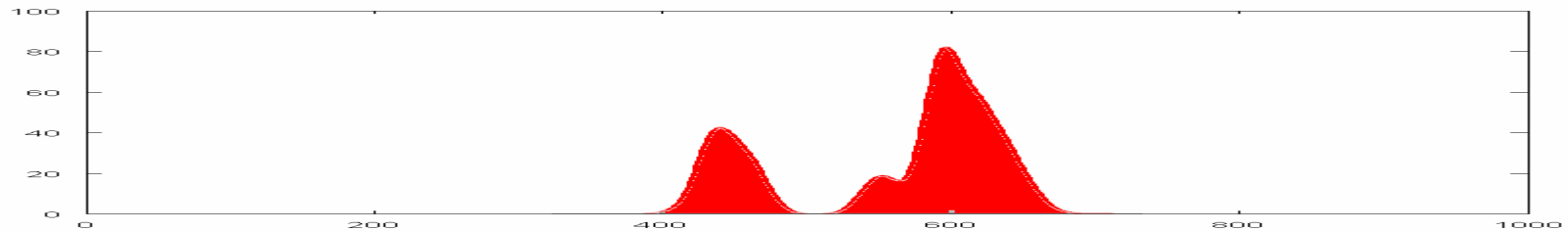


## Príklad signálu - odozva tretieho detektoru



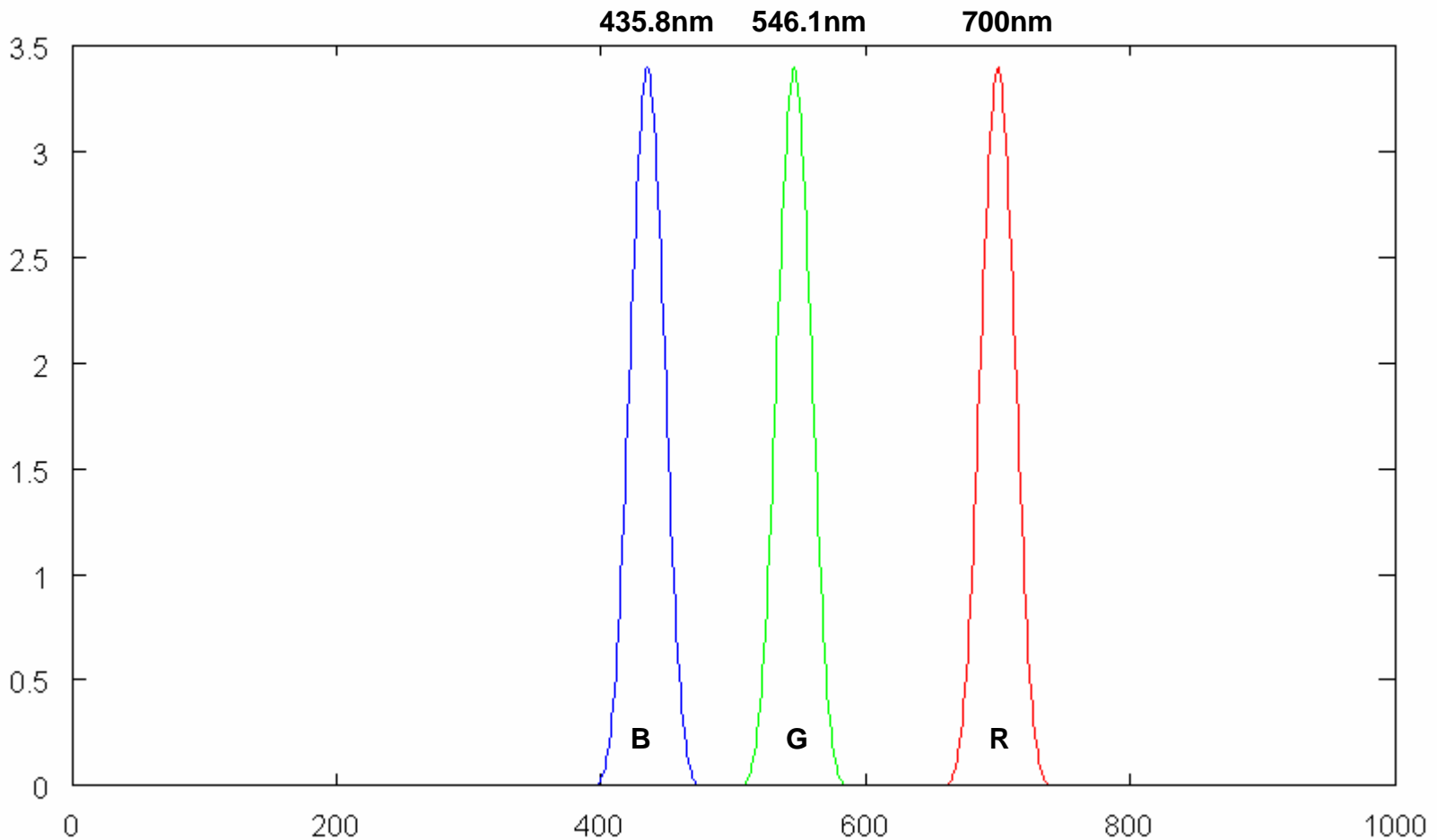
# Farba

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



# Farba

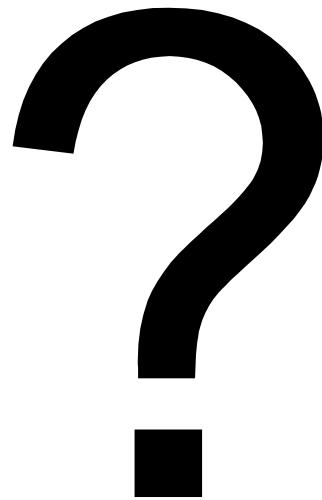
Príklad signálu – to využíva monitor počítača, ktorý generuje len tri vlnové dĺžky a túto farbu nasimuluje oku úplne inou distribúciou vlnových dĺžok:





# Farba

Príklad signálu – vnem je rovnaký a je to:



# Farba

Príklad signálu – vnem je rovnaký a je to šedá

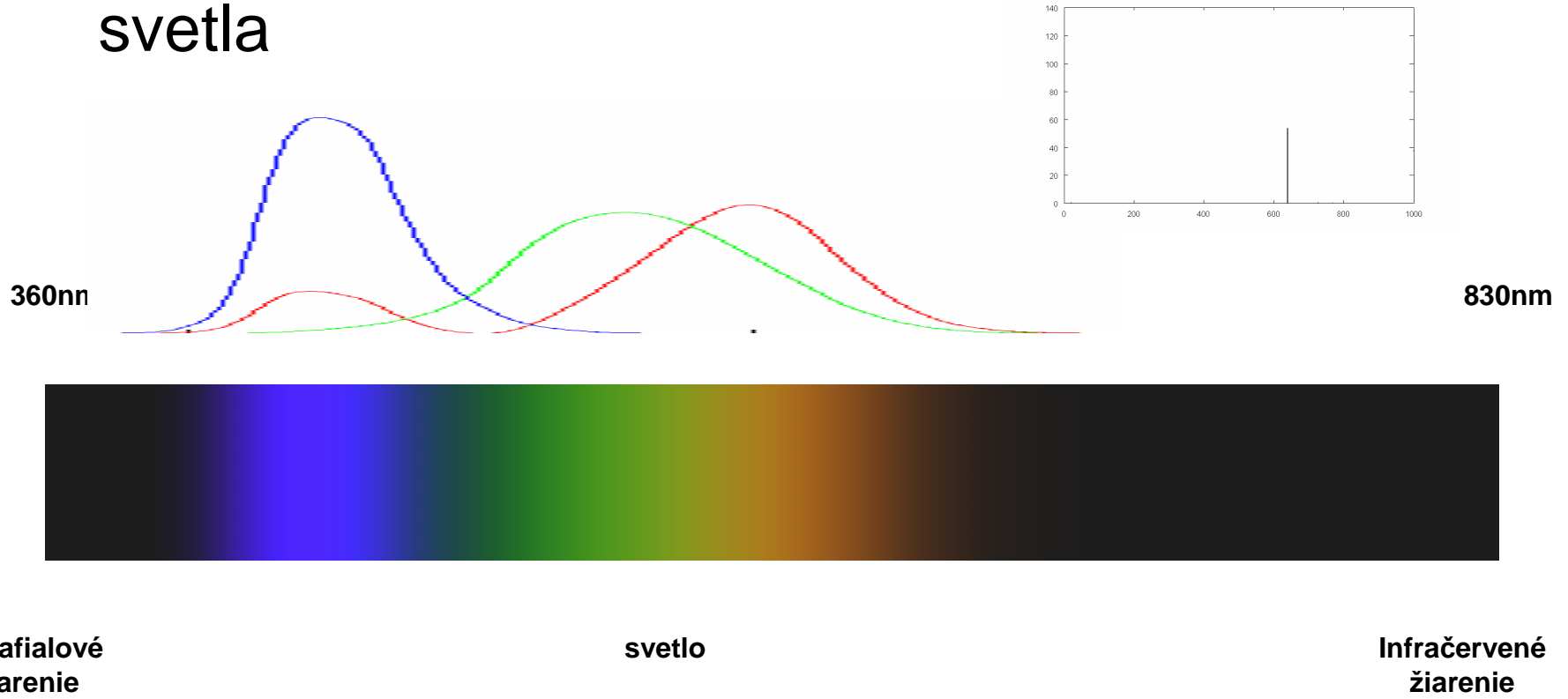


# Farba

- 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
- Optikou oka je ako-tak zabezpečené, že dopadajú na základe odrazu od určitého bodu na pozorovanom predmete
- Farba vyjadruje ako daný predmet odráža svetlo

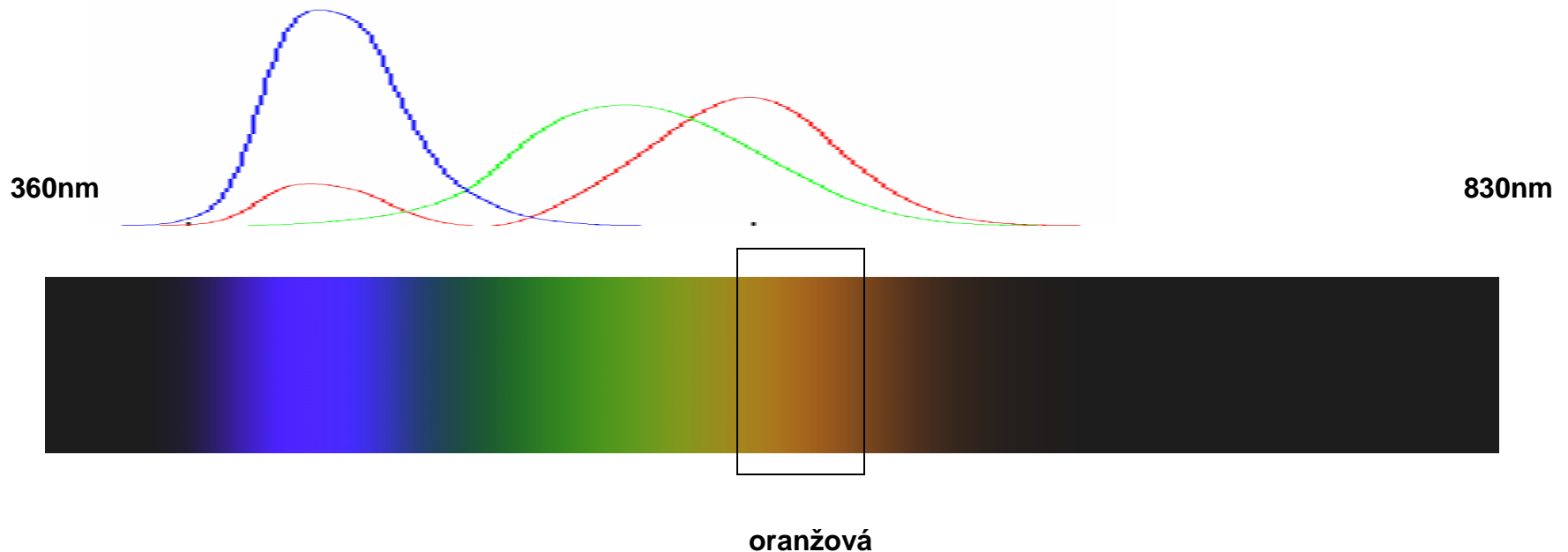
# Dúha

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



# 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 <  $c_1$  &&  $c_2$  < 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
- Náš obrázok je umelo vytvorený, musíme  
preto pozrieť na základe akej dohody o  
hraniciach bol vytvorený

# Norma

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

[http://en.wikipedia.org/wiki/Variations\\_of\\_orange](http://en.wikipedia.org/wiki/Variations_of_orange)

**Orange (web colour)**  
— Colour coordinates —

**Hex triplet** #FFA500

**sRGB<sup>B</sup>** (r, g, b) (255, 165, 0)

**HSV** (h, s, v) (39°, 100%, 100%)

**Source** [CSSX11/SVG<sup>\[2\]</sup>](#)

B: Normalized to [0–255] (byte)

**Tangerine**  
— Colour coordinates —

**Hex triplet** #F28500

**RGB<sup>B</sup>** (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<sup>B</sup>** (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<sup>B</sup>** (r, g, b) (255, 140, 0)

**HSV** (h, s, v) (34°, 100%, 100%)

**Source** [X11/SVG<sup>\[2\]</sup>](#)

B: Normalized to [0–255] (byte)

**Carrot orange**  
— Colour coordinates —

**Hex triplet** #ED9121

**RGB<sup>B</sup>** (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<sup>B</sup>** (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<sup>B</sup>** (r, g, b) (204, 85, 0)

**HSV** (h, s, v) (25°, 100%, 80%)

**Source** [University of Texas at Austin<sup>\[7\]</sup>](#)

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>)



# Program

```
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];
```

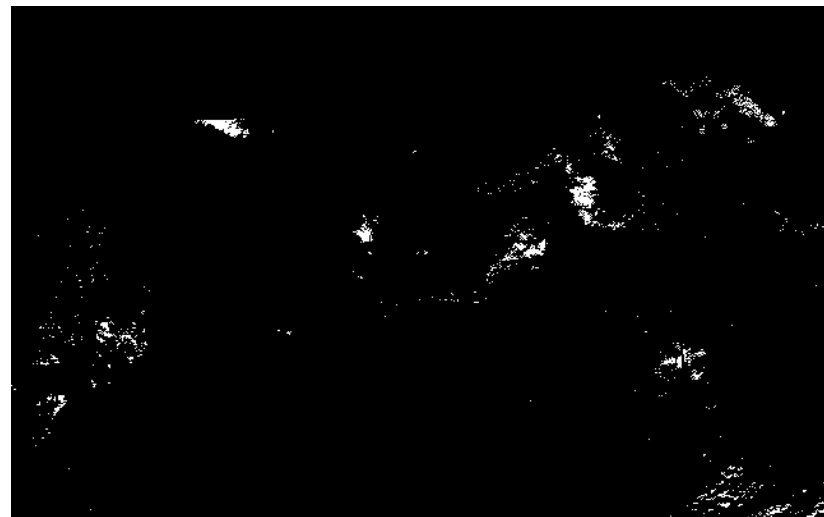
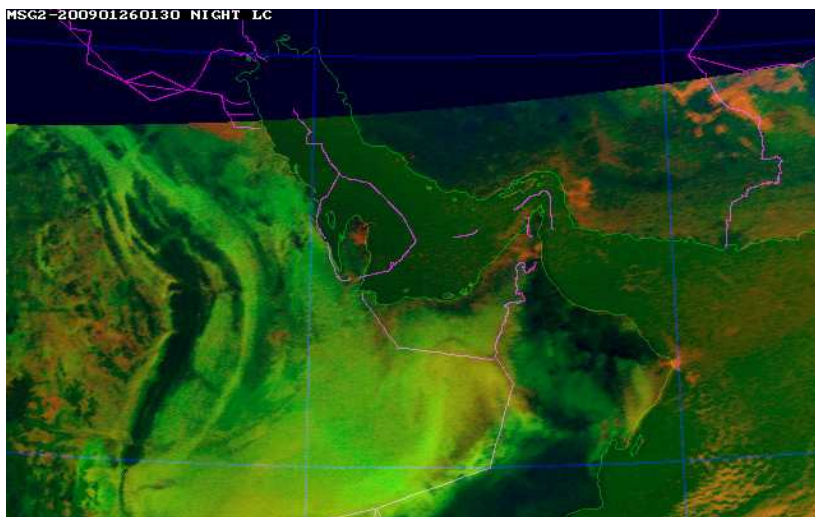
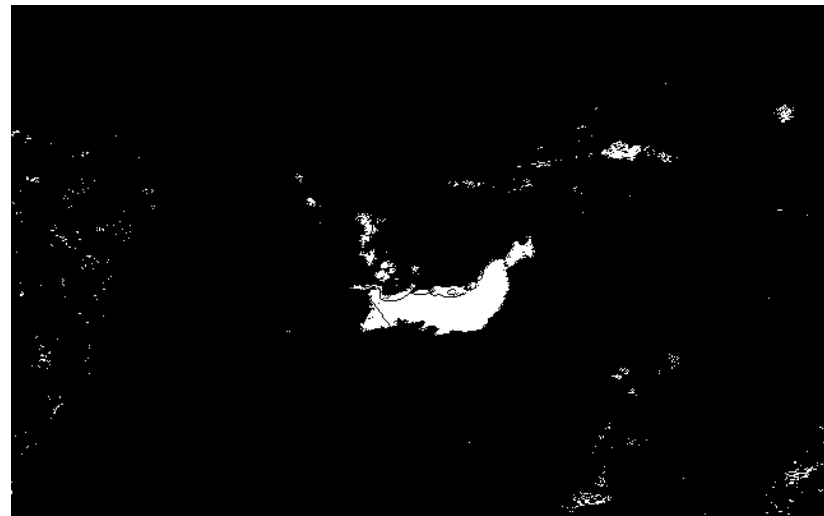
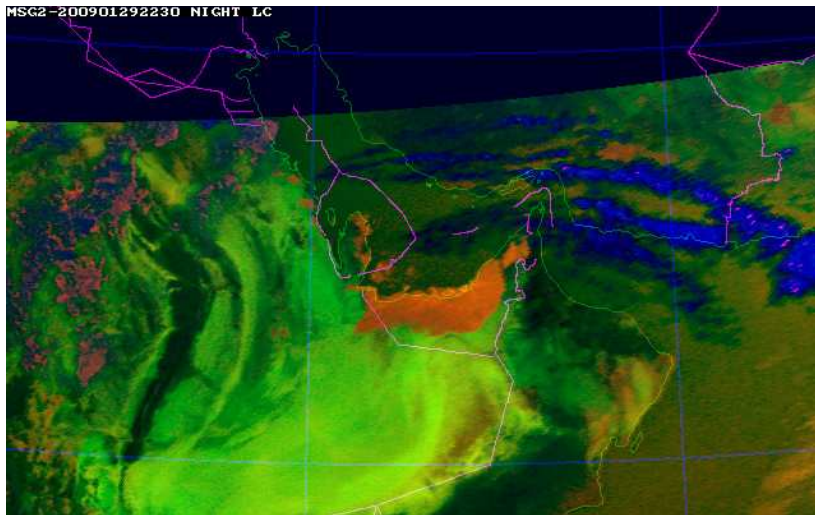
```
        endif
```

```
    end
```

```
end
```

```
jpgwrite([ "org-" fil ],img(:,:,1),img(:,:,2),img(:,:,3),100);
```

# Výsledky



Ďakujem za pozornosť!

Andrej Lúčný

[www.microstep-mis.com/~andy](http://www.microstep-mis.com/~andy)