

## 1. Segmentare imagini color (1): conversii intre modele de culoare si construirea histogramelor de culoare

Scop: implementarea unei metode de transformare a componentelor de culoare ale unei imagini color din modelul RGB24 (Red Green Blue) in modelul HSI (Hue Saturation Intensity), model invariant la variatii de luminozitate; calculul si afisarea histogramelor de culoare ale componentelor H, S si V.

### 1.1. Transformarea RGB $\Rightarrow$ HSV

H = Hue (culoare), S = Saturation (saturare), V = Value (intensitate)

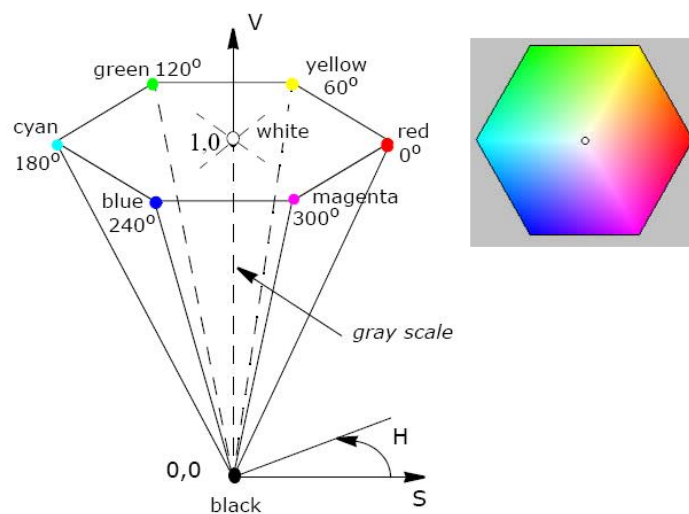


Fig. 1.1. Modelul (spațiul de culoare HSI).

Ecuatiile de transformare din componentele RGB în HSV sunt [2]:

```

r = R/255; // r : componenta R normalizata
g = G/255; // g : componenta G normalizata
b = B/255; // b : componenta B normalizata
// Atentie declarati toate variabilele pe care le folositi de tip float
// Daca ati declarat R de tip uchar, trebuie sa faceti cast: r = (float)R/255 !!!
    
```

```

M = max (r, g, b);
m = min (r, g, b);
C = M - m;
    
```

Value:

$$V = M;$$

Saturation:

```

If (C)
    S = C / V;
Else // grayscale
    S = 0;
    
```

Hue:

```

If (C) {
    if (M == r) H = 60 * (g - b) / C;
    if (M == g) H = 120 + 60 * (b - r) / C;
    }
    
```

```

    if (M == b) H = 240 + 60 * (r - g) / C;
    }
Else // grayscale
    H = 0;
If (H < 0)
    H = H + 360;

```

Valorile pt. H, S si V calculate cu formulele de mai sus vor avea următoarele domenii de valori:

H = 0 .. 360  
 S = 0 .. 1  
 V = 0 .. 1

Aceste valori se normalizează (scalează) în intervalul 0 .. 255 pt. a reprezenta fiecare componenta de culoare ca și o imagine cu 8 biți/pixel (de tip CV\_8UC1):

H\_norm = H\*255/360  
 S\_norm = S\*255  
 V\_norm=V\*255

## 1.2. Mersul lucrării

1. Se vor calcula histogramele componentelor de culoare normalizate (0 .. 255) ale H, S, V, Se vor afisa aceste histograme. Pt. afisare exista functia predefinita (*Functions.cpp*)

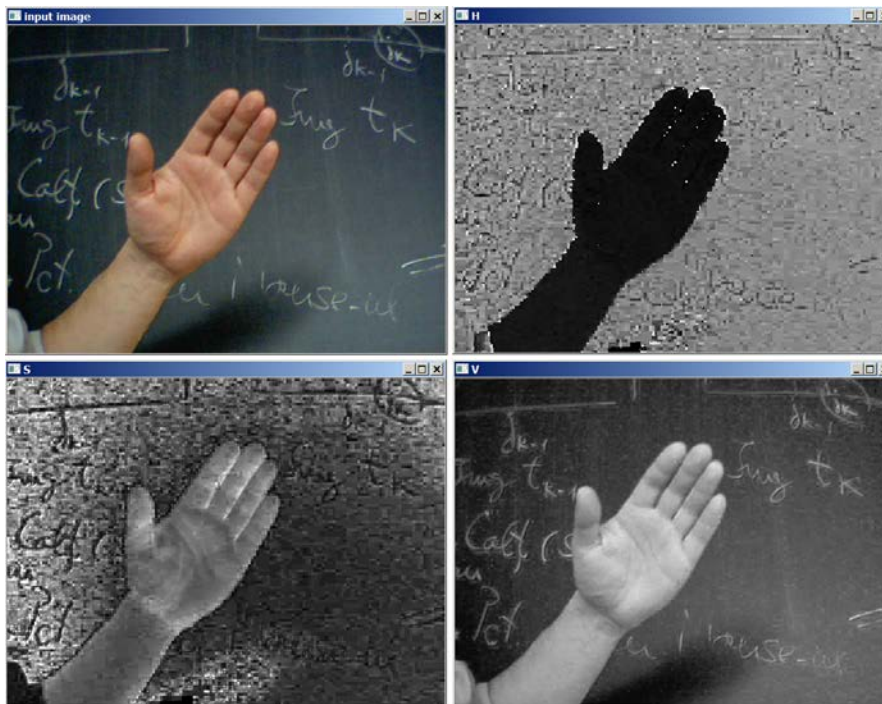
showHistogram (nume\_fereastră, hist, hist\_cols, hist\_height, true)

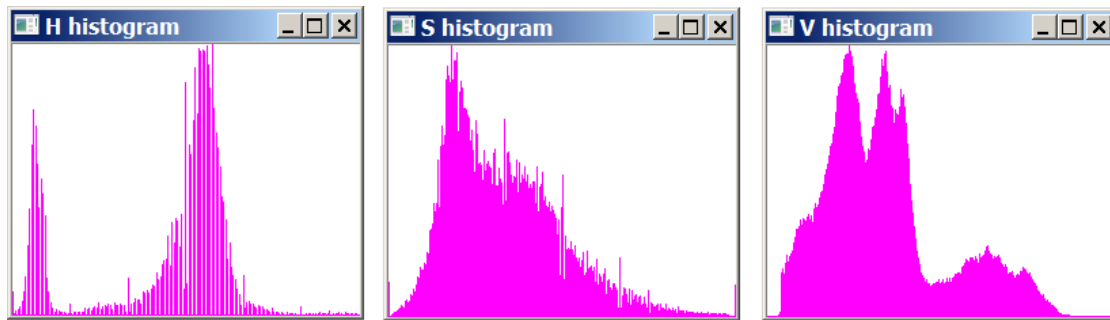
unde:

*hist* - vector de tip int in care se calculeaza valorile histogramei

*hist\_cols* – lungimea vectorului histograma (256)

*hist\_height* – inaltimea ferestrei in care se afiseaza histograma





2. Se va face o analiza a histogramelor H, S, V pentru toate imaginile de test si se va decide care componenta de culoare este cea mai potrivita pentru segmentarea mainii.
3. Se va binariza matricea componentei de culoare alese pt. segmentare cu un prag arbitrar ales astfel incat sa se separe cat mai bine pixelii de mana fata de cei de fundal si se va afisa rezultatul (fundalul cu negru, mana cu alb).
4. Calculati un prag de binarizare automat pentru componenta de culoare aleasa folosind implementarea algoritmului din capitolul 8.5 ( laboratorul 8 de la PI) si binarizati componenta respectiva cu acest prag. Comparati rezultatul cu cel de la pct. 3. Testati binarizarea pe toate imaginile de test din arhiva *img.zip*.

### Bibliografie

- [1] Intel, Color models, <https://software.intel.com/en-us/node/503873>.
- [2] Open Computer vision Library, Reference guide, cvtColor() function, [http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous\\_transformations.html#cvtColor](http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cvtColor)