

Capitolul 6

Lumină (a)cromatică

Culoarea unui obiect depinde nu numai de obiectul în sine ci și de sursa de lumină care îl luminează, de culoarea zonelor adiacente și de sistemul vizual uman.

Unele obiecte reflectă lumina (de ex. un perete, o foaie de hârtie) iar altele o transmit (de ex. o folie de celofan, sticla).

6.1 Lumină acromatică

Senzațiile vizuale acromatice pot fi descrise utilizând doar culorile negru, gri și alb.

Definiția 6.1

Lumină acromatică este ceea ce se poate observa pe un ecran TV sau monitor A/N¹. ■

Lumină acromatică are un singur atribut : *cantitatea*.

Definiția 6.2

Cantitatea de lumină poate fi descrisă prin conceptul fizic de *energie* (caz în care utilizăm termenii *intensitate* și *luminanță*) sau prin conceptul psihologic de *intensitate percepută* (caz în care utilizăm termenul de *strălucire*²). ■

Diverselor niveluri de intensitate li se asociază scalari : 0 (reprezentând culoarea neagră) și 1 (reprezentând culoarea albă). Nivelurile între 0 și 1 reprezintă nuanțe de gri.

6.1.1 Selectarea intensităților

În lucrul cu intensitățile luminii acromatice se pun 2 probleme :

¹alb/negru

²*brightness* în engleză

Problema 6.1

Dacă dorim să avem $n + 1$ niveluri de intensitate ($n \in \mathbb{N}$), cum le plasăm în intervalul $[0, 1]$? ■

Problema 6.2

În mod practic, de câte niveluri de intensitate este nevoie ? Câte niveluri de intensitate sunt suficiente ? ■

În cazul problemei 6.1, pag. 86 să presupunem că dorim să avem 256 ($n = 255$) niveluri de intensitate :

- Dacă am plasa 128 niveluri în $[0, 0.1]$ și alte 128 niveluri în $[0.9, 1]$ atunci trecerea de la intensitatea 0.1 la 0.9 ar apărea discontinuă (spre deosebire de celelalte tranziții).
- Dacă le-am plasa uniform în intervalul $[0, 1]$ (la distanță de $\frac{1}{255}$) atunci această alegere nu ar fi conformă cu o anumită caracteristică a ochiului uman :

Fapt 6.1 *Ochiul uman percepe nu atât valori absolute ale intensității cât rapoarte între intensități.*

Astfel trecerea de la intensitățile 0.1 și 0.5 la 0.11 și 0.55 este resimțită ca identică. În concluzie vom amplasa nivelurile de intensitate la distanțe logaritmice (și nu liniare) pentru a obține diferențe egale în strălucire, așa cum rezultă din figura 6.1, pag. 87.

Pentru obținerea celor $n + 1$ niveluri de intensitate procedăm astfel : fie I_0 nivelul de intensitate cel mai de jos posibil (din motive practice, pentru un CRT avem $I_0 \in [0.005, 0.025]$).

Alegem $I_1 = r \cdot I_0$, $I_2 = r \cdot I_1 = r^2 \cdot I_0$, ..., $I_j = r^j \cdot I_0$ ($0 \leq j \leq n$), ..., $I_n = r^n \cdot I_0 = 1$. Deci $r = \left(\frac{1}{I_0}\right)^{\frac{1}{n}}$ și $I_j = r^j \cdot I_0 = I_0^{1-\frac{j}{n}} = I_0^{\frac{n-j}{n}}$, $0 \leq j \leq n$.

Exemplul 6.1

Dacă $n = 3$, $I_0 = \frac{1}{8}$ atunci $r = 2$ și avem nivelurile de intensitate : $\{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$. ■

Definiția 6.3

Se numește *domeniu dinamic*³ valoarea $\frac{1}{I_0}$ (raportul dintre intensitatea maximă și minimă). ■

Observația 6.1

Din punct de vedere practic valoarea unui pixel și valoarea intensității sale calculată conform ecuației $I_j = r^j \cdot I_0$, $0 \leq j \leq n$ diferă. Se procedează astfel :

1. Intensitatea luminii pe ecranul unui CRT depinde de energia unei unde de electroni care stimulează niște particule de fosfor. Practic avem $I = k_1 \cdot N^\gamma$, k_1, γ constante și $\gamma \in [2.2, 2.5]$.

³ *dynamic range* în engleză

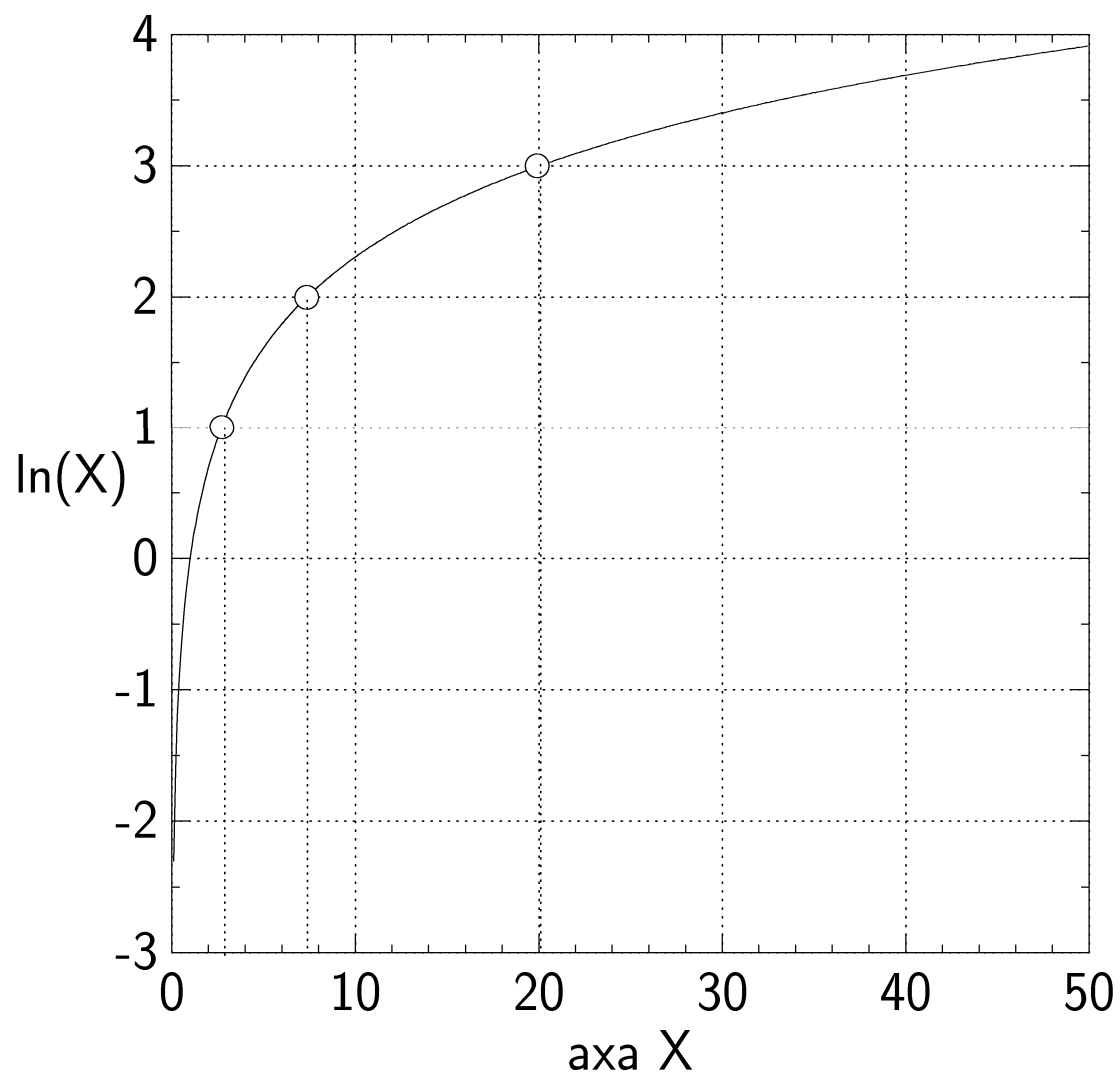


Figura 6.1: Amplasarea nivelurilor de intensitate

2. N este proporțională cu valoarea V a unui pixel și deci avem $I = K \cdot V^\gamma \Leftrightarrow V = \left(\frac{I}{K}\right)^{\frac{1}{\gamma}}$.

3. Să presupunem că dorim să determinăm valoarea V a unui pixel astfel încât aceasta să creeze intensitatea I . Se procedează astfel :

- Determinăm j astfel încât intensitatea I_j este cea mai apropiată de I : $I = I_j = r^j \cdot I_0 \Rightarrow j = \left\lceil \log_r \left(\frac{I}{I_0} \right) \right\rceil$.
- Deoarece $V = \left(\frac{I}{K}\right)^{\frac{1}{\gamma}}$ vom avea $V_j = \left[\left(\frac{I_j}{K}\right)^{\frac{1}{\gamma}}\right]$.

■

În cazul problemei 6.2, pag. 86 prin termenul de *suficient* ne gândim la numărul de intensități necesar pentru ca reproducerea unei imagini A/N (de ex. o fotografie A/N) să aibă un aspect *continuu* (i.e., între 2 pixeli vecini să nu existe diferențe mari de intensitate). Aspectul continuu este posibil doar pentru $r = 1.01$. S-a putut observa că dacă $r < 1.01$ ochiul nu poate distinge între intensitățile I_j și I_{j+1} . Deci, din ecuația $r = \left(\frac{1}{I_0}\right)^{\frac{1}{n}}$ obținem $r^n = \frac{1}{I_0} \Rightarrow n = \log_r \left(\frac{1}{I_0}\right)$ și deci $n = \log_{1.01} \left(\frac{1}{I_0}\right)$.

Câteva din valorile $\frac{1}{I_0}$ și n se regăsesc în tabelul 6.1, pag. 88.

	$\frac{1}{I_0}$	n
CRT	50 – 200	400 – 530
hârtie fotografică	100	465
hârtie A/N	100	465
hârtie culori	50	400
ziar A/N	10	234

Tabelul 6.1: Valori pentru $\frac{1}{I_0}$ și n

6.1.2 Aproximarea intensităților prin autotipie

Problema 6.3

În cazul ecranelor (sau al dispozitivelor de tipărire) nu putem avea decât un număr limitat de intensități (2 pentru ecrane monocrome sau 4, 8 dacă avem ecrane rastru cu 2, 3 biți/pixel). Cum putem obține, în aceste cazuri, niveluri de intensitate suplimentare ?

■

Pentru rezolvarea problemei 6.3, pag. 88 ne folosim de faptul că *ochiul realizează o integrare spațială* : dacă privim o zonă foarte mică de la o distanță suficient de mare atunci ochiul percepe o intensitate medie (și nu intensitatea fiecărui pixel în parte).

Tehnica de *autotipie*⁴ constă în simularea intensității prin cercuri negre de arie direct proporțională cu $1 - I$, unde I reprezintă valoarea intensității din fotografia originală. Aceasta tehnică se folosește, în general, pentru tipărirea fotografiilor A/N.

Dispozitivele de ieșire grafice pot aproxima cercurile de arie variabilă folosite în autotipie astfel : o zonă de 2×2 pixeli monocromi poate simula 5 niveluri de intensități diferite, cu prețul reducerii la jumătate a rezoluției spațiale⁵ pe fiecare axă (vezi figura 6.2, pag. 89).

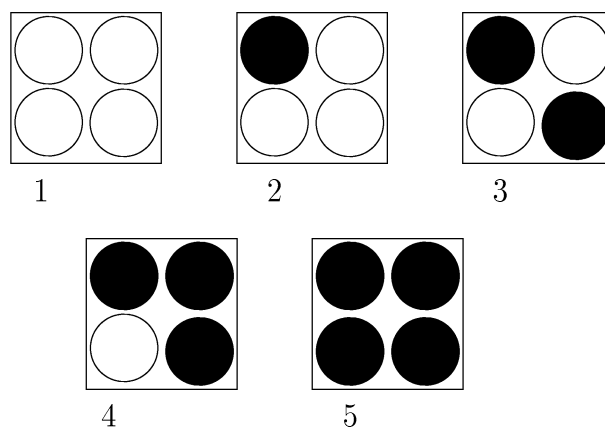


Figura 6.2: Simulare intensități

În general avem următoarea proprietate :

Proprietatea 6.1

Un grup de $n \times n$ pixeli monocromi poate furniza $n^2 + 1$ niveluri de intensitate. ■

Exemplul 6.2

O zonă de 3×3 pixeli monocromi poate simula 10 intensități : vezi figura 6.3, pag. 90.

Putem sintetiza aceste niveluri de intensitate într-o matrice de oscilație⁶ : $\begin{pmatrix} 6 & 8 & 4 \\ 1 & 0 & 3 \\ 5 & 2 & 7 \end{pmatrix}$, unde nivelul de intensitate I se obține setând toți pixelii cu valori mai mici decât I . ■

Putem avea mai multe modele pentru zonele de $n \times n$ pixeli utilizate în autotipie, dar aceste modele trebuie să satisfacă următoarele reguli :

- Un anumit model nu trebuie să introducă noi obiecte vizuale în zone de intensități egale. De exemplu, dacă în figura 6.3, pag. 90 în loc de modelul dat pentru 3 am fi utilizat modelul 3' am produce apariția unei linii orizontale în orice zonă a imaginii inițiale colorată uniform cu pixeli de intensitate 3.

⁴*halftoning* în engleză

⁵i.e., dacă o imagine era reprezentată pe $n \times n$ pixeli, fiecare pixel putând avea 5 intensități, în cazul când avem un ecran monocrom, pentru a avea o imagine la fel de detaliată avem nevoie de $2n \times 2n$ pixeli

⁶*Dither matrix* în engleză

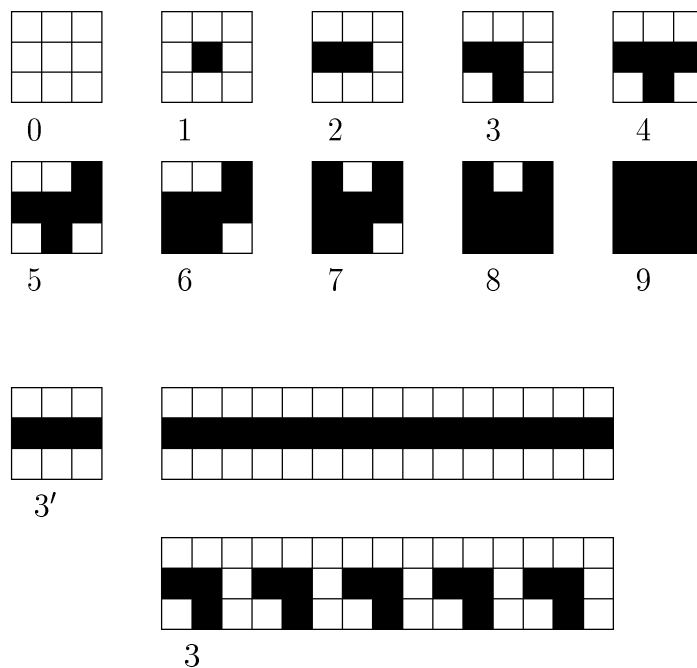


Figura 6.3: 10 niveluri de intensitate

- Modelele trebuie să alcătuiască o secvență crescătoare astfel încât dacă un pixel era intensificat pentru simularea nivelului de intensitate j atunci va rămâne intensificat și pentru toate nivelurile de intensitate $k > j$.
- Modelele trebuie să *crească* dinspre centru înspre exterior pentru a simula efectul de creștere a mărimii punctului.
- Pixelii setați (1) trebuie să fie grupați. Nu se admit pixeli 1 răsfireți.

Aproximarea diferitelor niveluri de intensitate nu se limitează doar la ecrane monocrome. În cazul unui ecran cu 2 biți pe pixel (i.e., 4 niveluri de intensitate pentru un pixel), dacă utilizăm zone de 2×2 pixeli, se pot simula 13 niveluri de intensitate (bineînțeleas cu prețul reducerii rezoluției spațiale) : vezi figura 6.4, pag. 91.

6.2 Lumină cromatică

Percepția culorii implică 3 valori : *nuanță*⁷, *saturație*⁸, *luminozitate*⁹. Nuanța distinge familiile de culori : avem nuanțe de roșu, verde, albastru, etc. Saturația se referă la distanța culorii

⁷ *Hue* în engleză

⁸ *Saturation* în engleză

⁹ *Lightness* în engleză

0 0	1 0	1 0	1 1	1 1	2 1
0 0	0 0	0 1	0 1	1 1	1 1
0	1	2	3	4	5

2 1	2 2	2 2	3 2	3 2	3 3	3 3
1 2	1 2	2 2	2 2	2 3	2 3	3 3
6	7	8	9	10	11	12

Figura 6.4: 13 niveluri de intensitate

respective față de culoarea gri care are aceeași intensitate (de ex., culoarea roșie este puternic saturată iar cea roz este relativ nesaturată, adică mai apropiată de culoarea gri de aceeași intensitate). Luminozitatea reprezintă intensitatea percepută a unui obiect care reflectă lumina. Strălucirea¹⁰ se referă la intensitatea percepută a unui obiect care emite lumină (de ex., un bec, soarele, CRT).

Există mai multe modalități de specificare și de măsurare a culorilor :

- prin compararea culorii necunoscute cu un set de culori standard (modelul de culori Munsell¹¹),
- în modelul artistic culoarea este specificată prin 3 parametri : nuanță¹², umbrire¹³, ton¹⁴, vezi figura 6.5, pag. 92. O *nuanță* rezultă prin adăugarea culorii alb unei culori pure, astfel scăzând saturația. O *umbrire* provine din adăugarea de culoare neagră unei culori pure, astfel scăzând luminozitatea. Un *ton* este obținut adăugând atât culoarea alb cât și negru unei culori pure. Obținem astfel culori de aceeași nuanță (hue) dar cu saturație și luminozitate variabile. Amestecarea culorilor alb/negru produce culori gri.

6.2.1 Modele de culori pentru grafica rastru

Definiția 6.4

Un *model de culori* constă într-un sistem (cartezian - în cazul RGB, cilindric - în cazul HSV) de coordonate 3D și o submulțime a \mathbf{R}^3 care conține toate culorile dintr-o anumită gamă de culori. ■

De exemplu, modelul RGB constă din cubul $[0, 1]^3$ definit într-un sistem de coordonate cartezian 3D.

¹⁰ *Brightness* în engleză

¹¹ *Munsell space* în engleză

¹² *Tint* în engleză

¹³ *Shade* în engleză

¹⁴ *Tone* în engleză

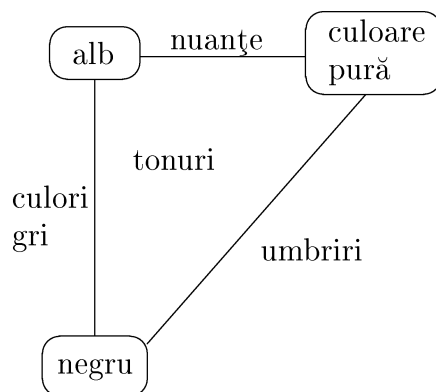


Figura 6.5: Modelul artistic

Modelul RGB

Modelul de culori RGB¹⁵ este utilizat pentru CRTs în culori precum și în grafica rastru. Spațiul de culori este cubul unitate $([0, 1] \times [0, 1] \times [0, 1])$ (vezi figura 6.6, pag. 93). O anumită culoare este reprezentată printr-un triplet de culori primare : (roșu, verde, albastru). Aceste culori primare sunt și *aditive* : fiecărei culori $C(r, g, b)$ îi corespunde bijectiv un punct din spațiul de culori RGB (punct care este dat de vectorul $r \cdot \vec{R} + g \cdot \vec{G} + b \cdot \vec{B}$, unde $\vec{R}, \vec{G}, \vec{B}$ sunt versori). Pe diagonala principală regăsim diverse nuanțe de gri.

Modelul HSV

Spre deosebire de alte modele de culori orientate spre hardware (de ex. RGB), modelul HSV¹⁶ este orientat spre utilizator, fiind apropiat de modelul artistic. Spațiul de culori HSV este o piramidă hexagonală într-un sistem de coordonate polar (vezi figura 6.7, pag. 94). Culorile gri sunt situate pe dreapta *black-white*.

O culoare este specificată în modelul HSV prin tripletul (h, s, v) , unde $h \in [0^\circ, 360^\circ)$, $s \in [0, 1]$ și $v \in [0, 1]$. Parametrul h (hue) semnifică nuanța culorii, s (saturation) specifică saturația culorii iar v (value) specifică strălucirea percepută a culorii. În modelul HSV, variind h alegem o culoare (dintr-o familie de culori), variind s modificăm saturația culorii (făcând-o să varieze între culoarea gri cu aceeași intensitate și culoarea pură) iar variind v variem contribuția culorii negre (variind astfel luminozitatea).

Corespondența dintre modelele RGB și HSV

Conversia culorilor din modelul RGB în HSV se efectuează proiectând cubul unitate RGB pe un plan perpendicular pe diagonala principală în punctul *white*.

¹⁵ *Red, Green, Blue* în engleză

¹⁶ *Hue, Saturation, Value* în engleză

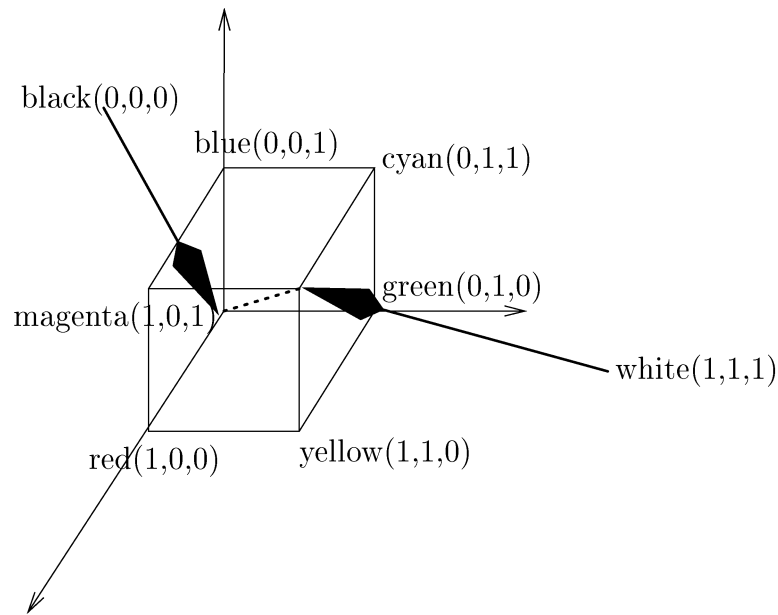


Figura 6.6: Modelul RGB

Algoritmul 12 (Conversia RGB în HSV)

Date de intrare : r, g, b : real

Date de ieșire : h, s, v : real

Metodă :

```

procedure RGBtoHSV(var h,s,v : real, r,g,b : real) {
  const int UNDEF = -1;
  real max, min, delta;

  max = valoarea_maxima(r,g,b);
  min = valoarea_minima(r,g,b);
  v = max;
  s = (max != 0.0) ? ((max - min)/max) : 0;
  if (s == 0) h = UNDEF;
  else {
    delta = max - min;
    if (delta == 0.0) { h = UNDEF; s = 0; v = 1; }
    if (max == 0) { h = s = UNDEF; v = 0; }
    if (r == max) h = (g - b) / delta;
    else if (g == max) h = (b - r) / delta + 2.0;
    else if (b == max) h = (r - g) / delta + 4.0;
    else {}
  }
}

```

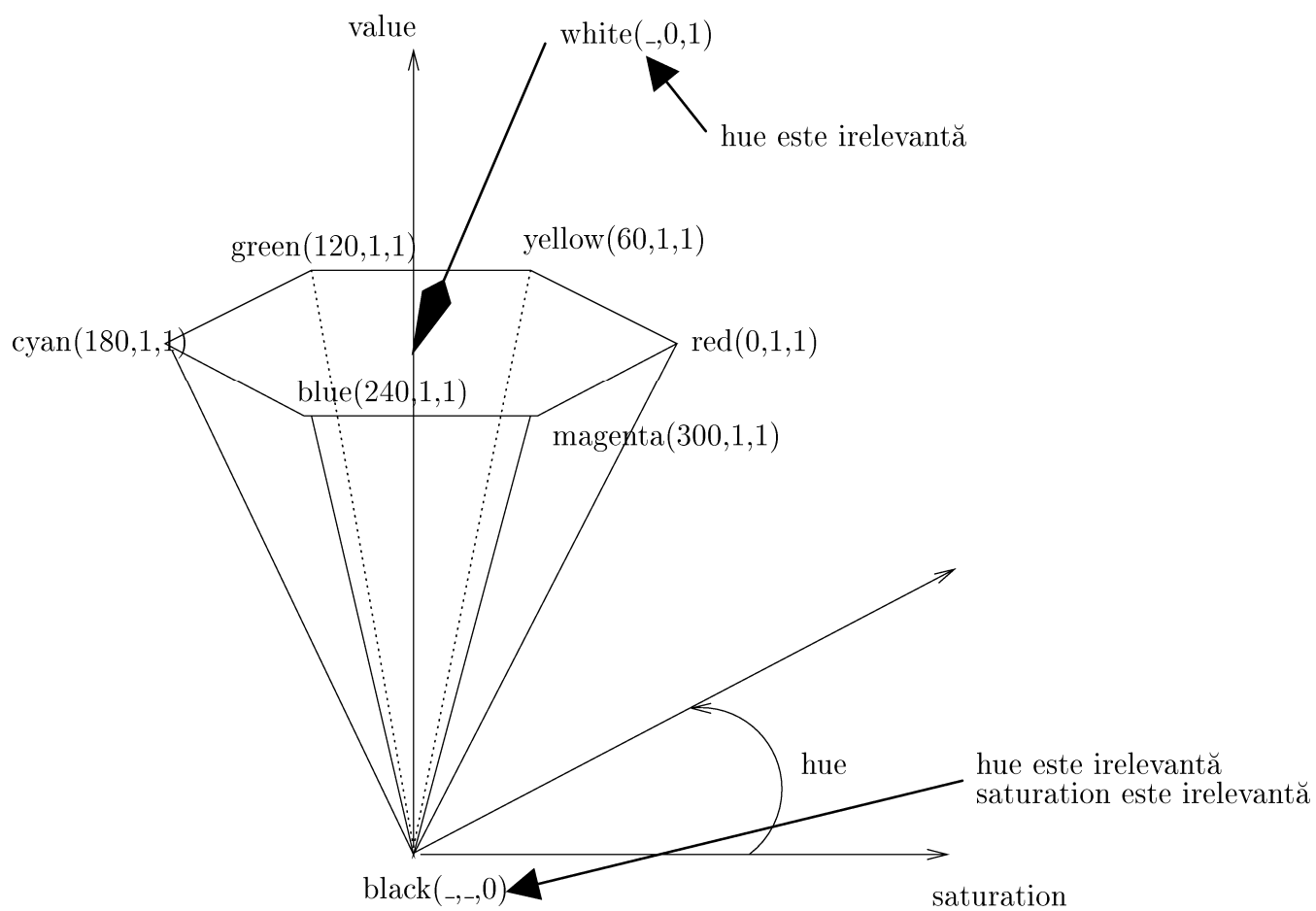


Figura 6.7: Modelul HSV

```

    h = h * 60;
    if (h < 0) h = h + 360;
}
}

```

Algoritmul 13 (Conversia HSV în RGB)

Date de intrare : $h, s, v : \text{real}$

Date de ieșire : $r, g, b : \text{real}$

Metodă :

```

procedure HSVtoRGB(var r,g,b : real, h,s,v : real) {
    const int UNDEF = -1;
    real f,p,q,t;
    int i;

    if (s == 0) {
        if (h == UNDEF) r = g = b = v;
        else printf("Eroare");
    }
    else {
        if (h == 360) h = 0;
        h = h / 60;
        i = floor(h);
        f = h - i;
        p = v * (1 - s);
        q = v * (1 - s * f);
        t = v * (1 - (s * (1 - f)));
        switch (i) {
            case 0 : r = v; g = t; b = p; break;
            case 1 : r = q; g = v; b = p; break;
            case 2 : r = p; g = v; b = t; break;
            case 3 : r = p; g = q; b = v; break;
            case 4 : r = t; g = p; b = v; break;
            case 5 : r = v; g = p; b = q; break;
        }
    }
}
}

```

6.2.2 Specificarea interactivă a culorii

În aplicațiile care utilizează culori pentru drepte, text, suprafețe, etc. există mai multe modalități (vezi figura 6.8, pag. 97) prin care se permite utilizatorului selectarea unei culori :

- Utilizatorul poate alege o culoare din mai multe mostre de culori (modelul indexat),
- Utilizatorul poate specifica coordonate R, G, B ,
- Utilizatorul poate specifica coordonate H, S, V .

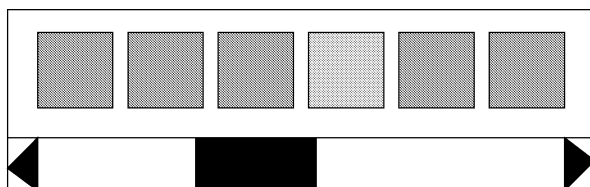
6.2.3 Utilizarea culorii în grafica pe calculator

Un *principiu fundamental* al utilizării culorii în diverse aplicații este următorul : utilizarea culorii trebuie să aibă un scop funcțional și să nu aibă un subînțeles.

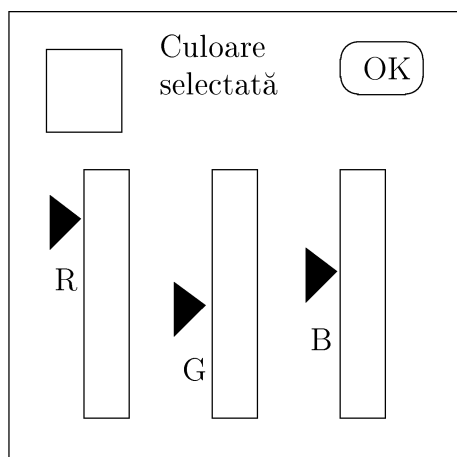
Reguli de estetică ale culorii

1. Culorile trebuie selectate conform unei anumite metode, de obicei prin traversarea unui drum continuu în modelul culorilor sau prin restricționarea culorilor la anumite planuri sau secțiuni hexagonale.
2. Nu se recomandă alegerea aleatoare a H, S .
3. Dacă pentru un grafic se utilizează câteva culori atunci se recomandă pentru fundal¹⁷ utilizarea unei culori care este complementară unei culori utilizate în grafic (culoarea complementara unei culori C este acea culoare care combinată cu C produce o culoare neutră : alb, gri sau negru. Pentru modelul RGB se obține prin simetrie față de centrul cubului iar pentru modelul HSV se obține prin simetrie față de centrul hexagonului). Dacă se utilizează mai multe culori atunci pentru fundal se recomandă culoarea gri.
4. Dacă alăturarea a 2 culori nu este armonioasă se recomandă utilizarea de chenare subțiri de culoare neagră.
5. Dacă dorim să asociem diverse înțelesuri culorilor atunci trebuie să luăm câteva precauții :
 - (a) Codurile culorilor pot avea înțelesuri implicite : de ex., *roșu* sugerează atenție, *eroare*, *verde* corect.
 - (b) Culorile strălucitoare denotă o importanță crescută asociată unui element.
 - (c) Două obiecte cu aceeași culoare pot fi interpretate ca având același cod (ceea ce nu e valabil întotdeauna).

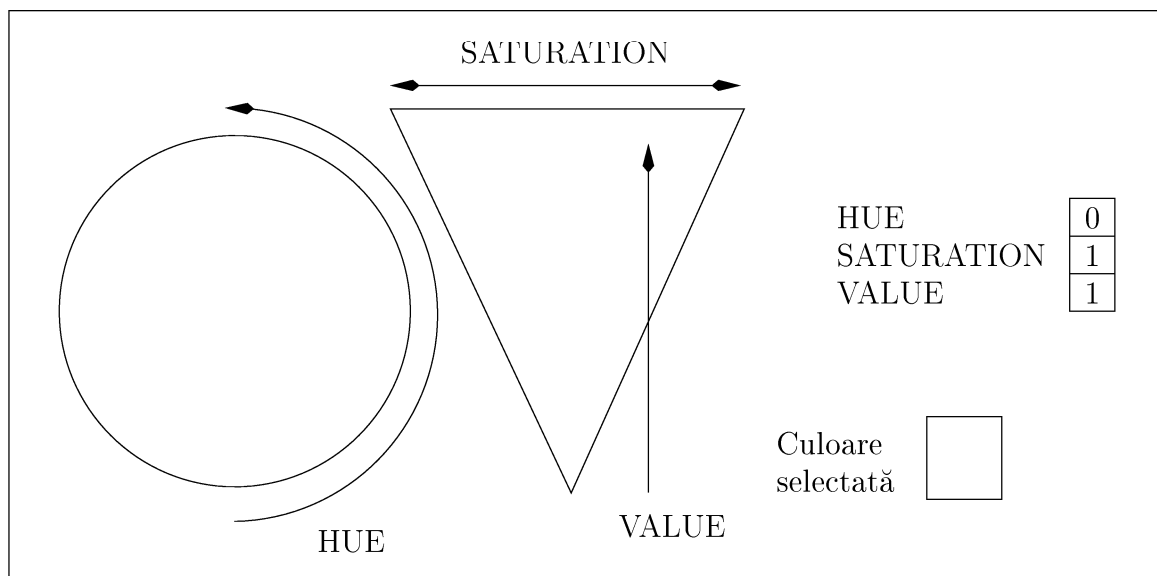
¹⁷ *Background* în engleză



Modelul indexat



Modelul RGB



Modelul HSV

Figura 6.8: Specificarea interactivă a culorii

6. Din regula 5, pag. 96 rezultă că utilizarea culorilor în meniuri, ferestre de dialog, chenare de ferestre trebuie să fie restricționată.
7. Detaliile fine (segmente de dreaptă, text) trebuie să difere de fundal nu doar în cromaticitate ci și în strălucire (în mod deosebit când se utilizează culori conținând albastru).
8. Combinații greșite de culori : albastru și negru (deoarece diferă foarte puțin în strălucire), galben și alb (idem). Combinații bune de culori : galben (pentru fundal) și negru (pentru text), text alb pe fundal albastru. Alte combinații greșite sunt : roșu și verde (când nu sunt strălucitoare pot crea probleme persoanelor care nu percep diferențe între cele două culori).
9. Ochiul nu poate distinge culoarea obiectelor foarte mici, deci nu se recomandă, într-o imagine colorarea pixelilor vecini cu culori distincte.
10. Culoarea percepută de către ochi a unei suprafețe este afectată de suprafața înconjurătoare. Se recomandă deci culori gri sau nesaturate pentru suprafețele înconjurătoare.
11. Culoarea unei suprafețe poate modifica mărimea sub care este percepută de către ochi (de ex., suprafețele colorate roșu sunt percepute ca fiind mai mari decât cele colorate verde).
12. Nu se recomandă utilizarea de suprafețe mari colorate cu o singură culoare.
13. Nu se recomandă colorarea a două suprafețe cu două culori de la capetele opuse ale spectrului (ROGVAIV) : de ex., roșu pentru fundal și albastru pentru prim-plan¹⁸. În acest caz roșu apare mai aproape iar albastru mai departe, invers față de ceea ce dorim.

¹⁸ *Foreground* în engleză