Travaux pratiques de Traitement d'Images sous MATLAB
Etude et applications du système IHS

Année 2016

1 Le codage RGB

Il consiste à représenter l'espace des couleurs à partir des trois rayonnements monochromatiques suivants :

1. le Rouge (R) de longueur d'onde 700 nm
2. le Vert (G) (Green en anglais) de longueur d'onde 546,1 nm
3. le Bleu (B) de longueur d'onde 435,8 nm

Le modèle RGB est le système le plus courant et le plus utilisé car il dérive de la technologie employée dans l'industrie de l'image et du numérique. Moniteurs à tube cathodique ou à cristaux liquides, appareils photos numériques, scanners, utilisent tous un système RGB basé sur le principe additif des trois couleurs primaires, rouge, vert et bleu. En codant chacune des composantes colorées sur un octet, on obtient 256 valeurs pour chaque couleur. Il est donc possible en théorie d'obtenir 256^3, soit 16.777.216 couleurs, c'est-à-dire beaucoup plus que l'œil humain n'est capable d'en discerner (environ 350000). Cette valeur reste cependant théorique, car les écrans ne permettent pas d'afficher un tel nombre de couleurs [1].

2 Espaces perceptuels

Comme leur nom l'indique, ces espaces se construisent à partir de trois paramètres servant à caractériser l'impression visuelle, à savoir :

1. l'intensité de la couleur (I)
2. sa teinte (en anglais, "hue") (H)
3. sa saturation (qui représente la pureté de la couleur) (S)

Pour cela, on définit un axe d'intensité I correspondant à tous les gris et allant du noir au blanc de référence. En se plaçant dans un plan perpendiculaire à cet axe et en tournant autour de cet axe, on parcourt toutes les teintes H. Enfin la distance d'un point à l'axe des intensités mesure la saturation de la couleur.

3 Le codage IHS et le passage de RGB à IHS

Ce codage permet de repérer les couleurs dans un espace perceptuel.
Le passage de RGB à IHS s'effectue en deux étapes, la premère est linéaire et la seconde non linéaire.

1. Calcul de : $\begin{pmatrix} I \\ \nu_1 \\ \nu_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \\ \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$
2. Calcul de : $\mathrm{H} = \mathrm{atan2}(\nu_2, \nu_1)$

1. d'après http ://e-cours.univ-paris1.fr/modules/uved/envcal/html/compositions-colorees/representations-couleur/systemes-materiels/rgb.html

2

et de S = $\sqrt{\nu_1^2 + \nu_2^2}$

Solution : fonction rgb2ihs.m

```
function image_ihs = rgb2ihs(image_rgb)
% Fonction effectuant la conversion d'une image RGB en IHS
%
M = [1/3 1/3 1/3; -1/2 -1/2 1; sqrt(3)/2 -sqrt(3)/2 0];
%
nlig = size(image_rgb,1);
ncol = size(image_rgb,2);
image_ihs = zeros(nlig,ncol,3);
%
for nolig=1 :nlig,
   for nocol=1 :ncol,
      image_ihs(nolig,nocol, :) = M * double(squeeze(image_rgb(nolig,nocol, :)));
   end
end
%
v1 = image_ihs( :, :,2);
v2 = image_ihs( :, :,3);
%
H = atan2(v2,v1);
S = sqrt(v1.^2 + v2.^2);
%
image_ihs( :, :,2) = H;
image_ihs( :, :,3) = S;
end
```

4 Programmation du codage IHS des images TRIMAGO et JPEG

Nous désirons coder en IHS les images au format TRIMAGO et au format JPEG. Nous procèderons en deux étapes :

1. Lecture d'une image au format TRIMAGO ou au format JPEG

   Solution : lecimage.m

   ```
   % Lecture d'une image au format TRIMAGO ou au format JPEG
   %
   NOMFIC = uigetfile('Images/*.*');
   suffixe = NOMFIC(find(NOMFIC=='.')+1 :length(NOMFIC));
   NOMFIC = ['Images/' NOMFIC];
   if strcmp(suffixe,'tri')
   lecshowtrimago2;
   else
   if strcmp(suffixe,'jpg')
   lecshowjpg2;
   else
   disp('ERREUR')
   return
   end
   ```

2. atan2(ν_2, ν_1) donne l'angle θ tel que $\tan(\theta) = \frac{\nu_2}{\nu_1}$, $\sin(\theta) = \frac{\nu_2}{\sqrt{\nu_1^2+\nu_2^2}}$, $\cos(\theta) = \frac{\nu_1}{\sqrt{\nu_1^2+\nu_2^2}}$ et $\theta \in [-\pi, \pi]$. Cette fonction diffère de l'arc tangente atan(y) qui donne l'angle θ tel que $\tan(\theta) = y$ et $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Ce serait une erreur que d'écrire H = atan($\frac{\nu_2}{\nu_1}$). Remarquez aussi qu'atan2(0,0) = 0. Donc si l'on considère un pixel de couleur noire, grise ou blanche (R=G=B), alors H = 0. La "teinte" du noir, du gris ou du blanc (H) serait alors celle du bleu qui vaut aussi 0, ce qui serait évidemment faux ! Il faut donc considérer que les couleurs noir, blanc ou gris n'ont pas de teinte

```
    end
```

2. Codage en IHS
   Solution : fonction rgb2ihs.m

Analysons le résultat du codage en IHS de la mire TV couleur : mireTV.jpg . Il est particulièrement intéressant d'afficher l'image des saturations (S) image_ihs(:, :,3). Que remarquons-nous ?

5 Le passage de IHS à RGB

Programmer le passage du codage IHS au codage RGB
Solution : fonction ihs2rgb.m

6 Calcul et visualisation de l'anneau des teintes

Nous nous intéressons à la composante des teintes (H). Pour cela nous allons construire l'anneau des teintes [3]
.
Solution anneau_teintes.m

```
% Construction de l'anneau des teintes
%
nlig = 256;
ncol = 256;
image_ihs = zeros(nlig,ncol,3);
cx = 128;
cy = 128;
Rmin = 30;
Rmax = 80;
%
for i = 1 :nlig
   for j=1 :ncol
      dist = sqrt((i-cx)^2 + (j-cy)^2);
      if Rmin <= dist && dist <= Rmax
         image_ihs(i,j,1) = 160;
         image_ihs(i,j,3) = 255;
         sintheta = (cx-i)/dist;
         costheta = (j-cy)/dist;
         angle = atan2(sintheta,costheta);
         image_ihs(i,j,2) = angle;
      else
         image_ihs(i,j,1) = 0;
         image_ihs(i,j,3)= 0;
      end
   end
end
%
image_rgb = ihs2rgb(image_ihs);
him = figure ('Name','Anneau des teintes','Units','pixels');
imshow(image_rgb/255);
```

3. Il serait plus correct de parler d'un anneau des couleurs où toutes les teintes sont présentes. Dans cet anneau, on trouve toutes les couleurs à l'exception du blanc, du noir et du gris. En effet, ces dernières n'ont pas de teinte

7 Ordre des teintes

Lorsqu'on tourne sur l'anneau dans le sens trigonométrique, on rencontre successivement les couleurs suivantes : bleu, violet, magenta, rouge, orange, jaune, vert, cyan, bleu, ...
Essayer d'attribuer à chaque nom de couleur un intervalle angulaire de la teinte de cette couleur.
Solution teinte.m

8 Rotation des teintes

On désire effectuer un changement global des teintes des pixels d'une image en effectuant une rotation des teintes. Pour calculer l'angle de cette rotation, on désigne une teinte à remplacer puis une teinte de remplacement.
Solution rot_teinte.m

```
% Rotation de teinte sur une image trimago ou jpeg
%
% Lecture d'une image trimago ou jpeg
%
lecimage;
%
% Codage IHS
image_ihs = rgb2ihs(im);
sauv = image_ihs;
%
disp('Désigner une couleur à remplacer');
[y1,x1] = ginput (1);
x1 = round(x1);
y1 = round(y1);
disp(['x1 = ', num2str(x1),' y1= ',num2str(y1)]);
angle1d = (image_ihs(x1,y1,2)/pi)*180;
disp(['Angle de la teinte à remplacer : ',num2str(angle1d),' degrés ** teinte : ',teinte(angle1d)]);
%
disp(' ');
%
disp('Désigner une couleur de remplacement');
[y2,x2] = ginput (1);
x2 = round(x2);
y2 = round(y2);
disp(['x2 = ', num2str(x2),' y2= ',num2str(y2)]);
angle2d = (image_ihs(x2,y2,2)/pi)*180;
disp(['Angle de la teinte de remplacement : ',num2str(angle2d),' degrés ** teinte : ',teinte(angle2d)]);
%
% Calcul de l'angle de rotation
theta0 = image_ihs(x1,y1,2) - image_ihs(x2,y2,2);
theta0d = num2str((theta0/pi)*180);
%
image_ihs( :, :,2) = image_ihs( :, :,2) - theta0;
image_rgb = ihs2rgb(image_ihs);
%
him = figure ('BackingStore','off','Name',['Rotation de teinte de ',num2str(theta0d), ' degres'],'Units','pixels');
imshow(image_rgb/255);
```

9 Extension des teintes au blanc

Bien que le blanc soit une couleur, il n'a pas de teinte. Le programme rot_teinte.m ne peut donc attribuer une couleur blanche (ou grise) à une autre couleur. Pour qu'il en soit ainsi, une solution va consister à changer le codage RGB en un codage quadridimensionnel RGBW (W représentant le blanc), puis à passer comme précédemment dans un espace perceptuel IH1H2S où l'on trouve cette fois-ci non plus un cercle mais une sphère des teintes. Dans cette sphère, la teinte blanche est maintenant présente. On peut alors par une rotation de la sphère (rotation définie par un angle de site et un angle d'azimut) changer une teinte donnée (celle du rouge par exemple) en une teinte blanche. Pour visualiser le résultat, il suffit de retrouver l'image RGB correspondante, en passant du code IH1H2S au code RGBW puis au code RGB.

9.1 Le passage de RGB à RGBW et à IH1H2S

Solution rgbw2ih1h2s.m

```
function image_ihs = rgbw2ih1h2s(image_rgb)
%
% Fonction effectuant la conversion d'une image RGBW en IH1H2S
% W = White (Blanc)
%
% Au départ, on a une image rgb : image_rgb
%
% Prétraitement : Passage de RGB à RGBW
% Pour cela, on retire tout le blanc possible de RGB et on le met dans W
R = image_rgb( :, :,1) ;
G = image_rgb( :, :,2) ;
B = image_rgb( :, :,3) ;
W = min(R,G) ;
image_rgb( :, :,4) = min(W,B) ;
%
for k = 1 :3
   image_rgb( :, :,k) = image_rgb( :, :,k) -image_rgb( :, :,4) ;
end
%
M = [1/3 1/3 1/3 1/3 ; -1/3 -1/3 -1/3 1 ; -1/2 -1/2 1 0 ; sqrt(3)/2 -sqrt(3)/2 0 0] ;
% (M (4,4) n'est pas orthogonale)
%
nlig = size(image_rgb,1) ;
ncol = size(image_rgb,2) ;
image_ihs = zeros(nlig,ncol,4) ;
%
for nolig=1 :nlig,
   for nocol=1 :ncol,
      image_ihs(nolig,nocol, :) = M * double(squeeze(image_rgb(nolig,nocol, :))) ;
   end
end
%
v1 = image_ihs( :, :,2) ;
v2 = image_ihs( :, :,3) ;
v3 = image_ihs( :, :,4) ;
%
H1 = atan2(v3,v2) ;
S = sqrt(v1.^2 + v2.^2 + v3.^2) ;
H2 = asin(v1./S) ;

%
```

```
image_ihs( :, :,2) = H1;
image_ihs( :, :,3) = H2;
image_ihs( :, :,4) = S;
end
```

9.2 Le passage de IH1H2S à RGB

Solution ih1h2s2rgb.m

9.3 Rotation de la sphère des teintes généralisées (blanc inclus)

Solution rot_teinte2.m

```
% Rotation de teinte sur une image trimago ou jpeg
% On utilise le codage IH1H2S
%
% Lecture d'une image trimago ou jpeg
%
lecimage;
%
% Codage IH1H2S
image_ihs = rgbw2ih1h2s(im);
%
disp('Désigner une couleur à remplacer');
[y1,x1] = ginput (1);
x1 = round(x1);
y1 = round(y1);
disp(['x1 = ', num2str(x1),' y1= ',num2str(y1)]);
%
psi1d = (image_ihs(x1,y1,2)/pi)*180; % psi1 en degres
theta1d = (image_ihs(x1,y1,3)/pi)*180; % theta1 en degres
disp(['Angle azimutal de la teinte à remplacer : ',num2str(psi1d),' degrés']);
disp(['Angle de site de la teinte à remplacer : ',num2str(theta1d),' degrés']);
%
disp(' ');
%
disp('Désigner une couleur de remplacement');
[y2,x2] = ginput (1);
x2 = round(x2);
y2 = round(y2);
disp(['x2 = ', num2str(x2),' y2= ',num2str(y2)]);
%
psi2d = (image_ihs(x2,y2,2)/pi)*180; % psi2 en degres
theta2d = (image_ihs(x2,y2,3)/pi)*180; % theta2 en degres
disp(['Angle azimutal de la teinte de remplacement : ',num2str(psi2d),' degres']);
disp(['Angle de site de la teinte de remplacement : ',num2str(theta2d),' degres']);
%
% Calcul des angles de rotation
psi = ??
psid = ?? %psi en degres
theta = ??
thetad = ?? %theta en degres
%
image_ihs( :, :,2) = ??
image_ihs( :, :,3) =??
```

```
% Codage RGB
image_rgb = ih1h2s2rgb(image_ihs);
image_rgb = image_rgb(:,:,1:3);
%
him = figure ('BackingStore','off','Name',['Rotation de teinte de ( ',num2str(psid),' , ',num2str(thetad), ' ) degres'],'Units','pixels');
imshow(image_rgb/255);
%image_rgb = image_rgb.*(image_rgb>=0) + (255-image_rgb).*(image_rgb>255);
%image(image_rgb/255);
```