

Travaux Pratiques Traitement Numérique du Signal

Alexis TREILLES
Téo FREJAVILLE

Mars 2023



Encadrant :
Marc Donias

Table des matières

1	Introduction	3
2	Introduction	3
3	Chroma-keying	3
3.1	Fusion des images <i>metro.jpg</i> et <i>people.jpg</i>	3
3.2	Fusions des images <i>foreground.jpg</i> et <i>background.jpg</i>	5
3.3	Fusions des images <i>foreground.jpg</i> et <i>background.jpg</i> avec représentation RGB	7
4	Segmentation des pastilles bleues	8
5	Changement de la teinte des pastilles	10
6	Conclusion	10
7	Annexe :	10
7.1	Changement de fond YCbCr	10
7.2	Changement de fond YCbCr	11
7.3	Changement de fond RGB	11
7.4	Création du masque pour les pastilles	12
7.5	Changement de couleur des pastilles	13

1 Introduction

2 Introduction

Dans le domaine du traitement d'images, la segmentation couleur se révèle être une technique fondamentale permettant de diviser une image en régions distinctes sur la base de critères colorimétriques. Cette méthode, appelée segmentation couleur, repose sur la partition de l'image en ensembles de pixels connectés, facilitant ainsi l'identification et l'analyse de régions spécifiques. Les applications de cette approche sont vastes, allant de l'incrustation d'objets ou de personnages dans une scène à la détection ou au suivi d'objets colorés, en passant par des manipulations comme le changement de teinte de certaines parties d'une image.

Parmi les applications emblématiques de la segmentation couleur se trouve le chroma-keying, également connu sous le nom d'incrustation en chrominance. Cette méthode consiste à fusionner deux images en couleurs en isolant et en remplaçant les pixels constituant l'arrière-plan d'une première image par ceux correspondant à une seconde image. Typiquement, dans le contexte du chroma-keying, les pixels de fond sont remplacés par ceux d'une image de référence, souvent réalisée sur un fond uni de couleur verte ou bleue (voir figure 1). Un exemple classique de cette technique est l'utilisation d'une carte météorologique en arrière-plan lors de la diffusion d'un bulletin météo, où le présentateur est filmé devant un écran de couleur uniforme.

3 Chroma-keying

3.1 Fusion des images *metro.jpg* et *people.jpg*

Nous fusionnerons les images 'metro.jpg' (arrière-plan) et 'people.jpg' (avant-plan) en utilisant la représentation YCbCr. Nous remplacerons le fond de l'image avant-plan par celui de l'image arrière-plan, en utilisant le bleu comme couleur de référence pour créer un masque de fusion.

Nous avons, dans un premier temps diviser une matrice en trois matrices unidimensionnelles R, G et B, chacune représentant une des composantes de couleur RGB de l'image. Puis nous avons calculé la luminance de l'image (Y) en combinant les composantes R, G et B avec des pondérations spécifiques à l'aide de l'équation suivante :

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Nous calculons la chrominance bleue (Cb) en utilisant la différence entre la composante bleue et la luminance, puis nous ajustons les valeurs pour qu'elles soient dans la plage de 0 à 255.

$$Cb = 0.564 * (B - Y) + 128$$

Pour finir nous avons calculé la chrominance rouge (Cr) de manière similaire à la chrominance bleue.

$$Cr = 0.713 * (R - Y) + 128$$

Une fois tous les calculs effectués nous avons défini un masque M en comparant les valeurs de la chrominance bleue avec un seuil de 140. Si la chrominance bleue est inférieure au seuil, M prend la valeur 1 sinon il prend 0. Par la suite on multiplie cette valeur par 255 afin d'avoir seulement 2 couleurs (blanc et noir):



Figure 1: Masque

Puis nous avons convolué le masque avec l'image *people.jpg* et nous avons obtenu :

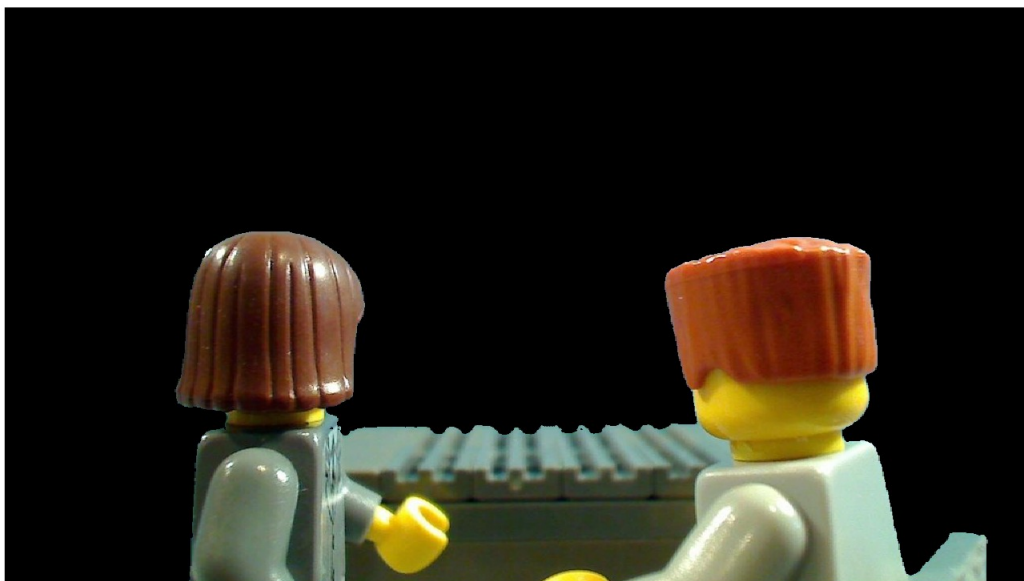


Figure 2: Convolution entre le premier plan et le masque

Ensuite nous avons utilisé le même procédé pour déterminer l'arrière plan, mais cette fois-ci en inversant le masque. Ainsi nous obtenons :

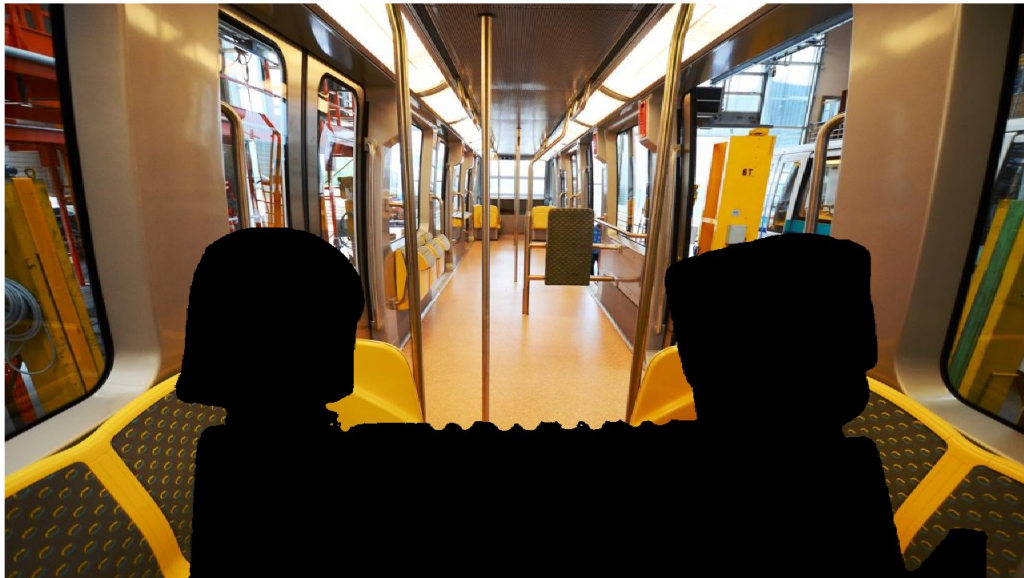


Figure 3: Convolution entre l'arrière plan et le masque inverse

Pour finir, nous avons sommé les deux images pour créer l'image finale. Nous obtenons :



Figure 4: Image finale

3.2 Fusions des images *foreground.jpg* et *background.jpg*

Nous fusionnerons les images 'background.jpg' (arrière-plan) et 'foreground.jpg' (avant-plan) en utilisant la représentation YCbCr. Notre objectif est de remplacer le ciel de l'image avant-plan par celui de l'image arrière-plan, en utilisant le bleu comme couleur de référence pour créer un masque de fusion.

La démarche afin de concevoir l'image finale est issue du même procédé que nous avons utilisé précédemment. Voici le masque que nous obtenons :

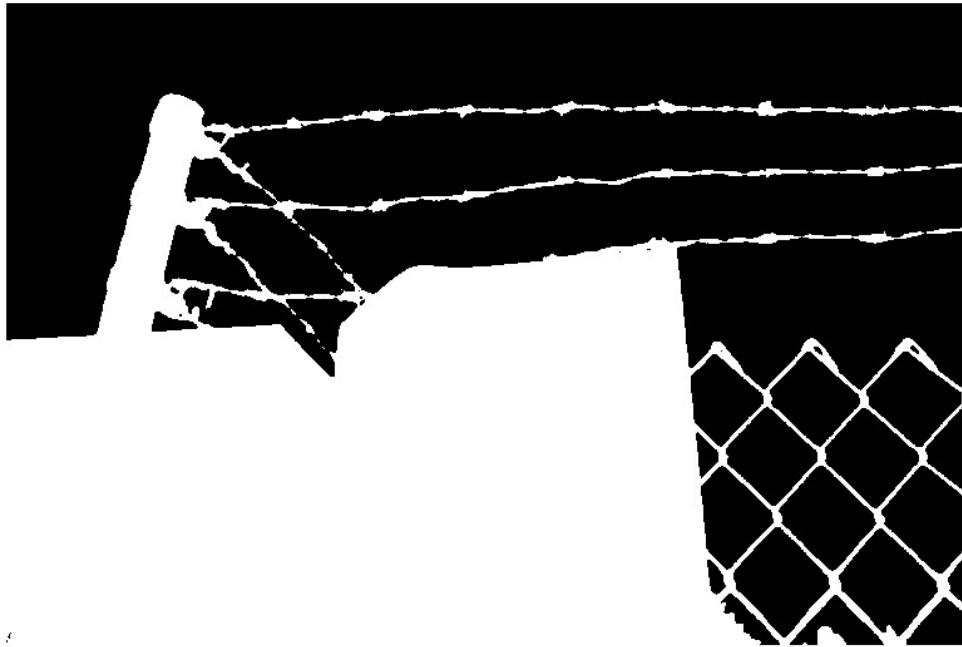


Figure 5: Masque

Par la suite nous avons sommé les deux images et nous trouvons l'image finale suivante :



Figure 6: Image finale

3.3 Fusions des images *foreground.jpg* et *background.jpg* avec représentation RGB

Nous allons essayer de fusionner les images en se basant uniquement sur la représentation RGB. Pour évaluer l'adéquation de la représentation YCbCr dans ce contexte, nous analyserons pourquoi la fusion peut être plus complexe avec la représentation RGB.

Nous avons, dans un premier temps diviser une matrice en trois matrices unidimensionnelles R, G et B, chacune représentant une des composantes de couleur RGB de l'image.

Par la suite nous avons défini un masque plus complexe où des seuil d'intensité de couleur sont mis sur les couleurs rouge, bleu et noir simultanément.

$$M = B > 80 \ \& \ G < 90 \ \& \ R < 90;$$

Ainsi nous obtenons le masque suivant :



Figure 7: Masque

Puis nous avons convolué le masque avec l'image *people.jpg*. Ensuite nous avons utilisé le même procédé pour déterminer l'arrière plan, mais cette fois-ci en inversant le masque. Pour finir, nous avons sommé les 2 images pour créer l'image finale. Nous obtenons :



Figure 8: Masque

Nous constatons une grosse erreur de convolution en bas à droite de l'image et quelques petit point de couleur le long du poteau à gauche. Nous avons ajusté les paramètre afin de les faire disparaître. Malheureusement, ces "taches" perciste toujours. Cela est du à certaine composante de couleur que nous n'avons pu enlevé avec le masque.

4 Segmentation des pastilles bleues

En pratique, cela nous donnera une plus grande flexibilité. Se baser sur un échantillon de référence est plus précis que se baser sur une couleur de référence comme précédemment. Pour cela, on va utiliser la distance de Mahalanobis qui s'exprime de la manière suivante :

$$D_{\text{Maha}}(\mathbf{y}) = (\mathbf{y} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}).$$

Avec :

1. \mathbf{y} un pixel,
2. $\boldsymbol{\mu}$ le vecteur moyen de notre échantillon,
3. $\boldsymbol{\Sigma}$ la matrice de covariance.

Cette distance est indicative ; elle représente la corrélation du pixel avec l'échantillon de référence. Ainsi, les pixels affichant une faible distance sont ceux qui sont très probablement ceux que l'on souhaite isoler.

On va ensuite définir le filtre en essayant différentes valeurs, dans notre cas on a :

$$M = 5 < D_{\text{Maha}}(\mathbf{y}) < 60$$

On obtient le masque suivant :



Figure 9: Masque

Une fois le masque appliqué à l'image on obtient le résultat suivant :

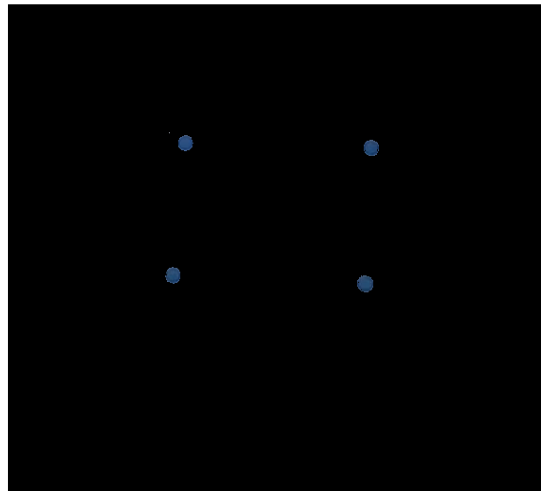


Figure 10: Résultat après application du filtre

On retrouve bien nos pastilles isolées du reste de l'image.

5 Changement de la teinte des pastilles

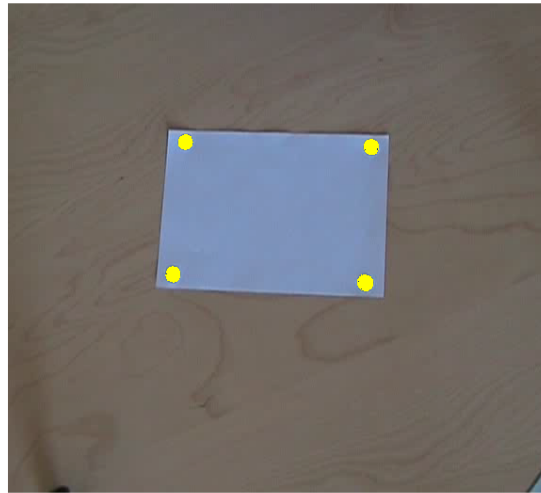


Figure 11: Résultat après changement de couleur

Ici on passe dans l'espace hsv, on change les couleurs uniquement où il y a le masque, on remet ensuite le fond d'origine.

6 Conclusion

Ce travail pratique en traitement numérique du signal a démontré l'efficacité de la segmentation couleur et en particulier de la méthode de chroma-keying pour isoler et manipuler des régions spécifiques d'une image. En utilisant la représentation YCbCr et la distance de Mahalanobis, nous avons pu isoler avec succès des éléments d'une image et changer leur teinte, montrant la polyvalence et la puissance des techniques de traitement d'images. Cette approche peut s'avérer précieuse dans divers domaines tels que la post-production vidéo, la reconnaissance d'objets et la réalité augmentée. Nos expériences ont confirmé l'importance de choisir des échantillons de référence appropriés pour les masques de segmentation, ce qui affecte directement la précision du résultat final. Bien que certaines limitations aient été observées, notamment la persistance de taches de couleur indésirables, les résultats obtenus sont prometteurs et ouvrent la voie à de futures améliorations et applications de ces techniques.

7 Annexe :

7.1 Changement de fond YCbCr

```
1 % superposition l go m tro :
2 clear, close all
3 m=double(imread('metro.jpg'));
4 %figure, imshow(m)
5
6
7 %figure, imshow(p)
8
9 p=double(imread('people.jpg'));
10 R=p(:,:,1);
11 G=p(:,:,2);
12 B=p(:,:,3);
13 Y=0.299*R+0.587*G+0.114*B;
14 Cb=0.564*(B-Y)+128;
```

```

15 Cr=0.713*(R-Y)+128;
16 L=(R+G+B)/3;
17
18 M=Cb<140;
19 %figure, imshow(uint8(M*255))
20 avant_plan=p.*M;
21 %figure, imshow(uint8(avant_plan))
22 arriere_plan=m.*(1-M);
23 %figure, imshow(uint8(arriere_plan))
24
25 image_finale=avant_plan+arriere_plan
26 %figure, imshow(uint8(image_finale))

```

7.2 Changement de fond YCbCr

```

1 %Background YCBcr
2 clear, close all
3 b=double(imread('background.jpg'));
4 %figure, imshow(m)
5
6 %figure, imshow(p)
7
8 f=double(imread('foreground.jpg'));
9 R=f(:, :, 1);
10 G=f(:, :, 2);
11 B=f(:, :, 3);
12 Y=0.299*R+0.587*G+0.114*B;
13 Cb=0.564*(B-Y)+128;
14 Cr=0.713*(R-Y)+128;
15 L=(R+G+B)/3;
16
17 M=Cb<140;
18 %figure, imshow(uint8(M*255))
19 avant_plan=f.*M;
20 %figure, imshow(uint8(avant_plan))
21 arriere_plan=b.*(1-M);
22 %figure, imshow(uint8(arriere_plan))
23
24 image_finale=avant_plan+arriere_plan
25 %figure, imshow(uint8(image_finale))

```

7.3 Changement de fond RGB

```

1 %Background RGB
2
3 clear, close all
4 f=double(imread('foreground.jpg'));
5 b=double(imread('background.jpg'));
6 R=f(:, :, 1);
7 G=f(:, :, 2);
8 B=f(:, :, 3);
9
10 M=B>80&G<90&R<90;
11 %figure, imshow(uint8(M*255))
12 avant_plan=b.*M;
13 %figure, imshow(uint8(avant_plan))
14 arriere_plan=f.*(1-M);
15 %figure, imshow(uint8(arriere_plan))
16 image_finale=avant_plan+arriere_plan

```

```
17 %figure, imshow(uint8(image_finale))
```

7.4 Création du masque pour les pastilles

```
1 % Pastilles
2 clear, close all
3
4
5 pastilles=double(imread('pastilles.png'))
6 figure, imshow(uint8(pastilles))
7
8 ptx= 182:196;
9 pty=142:155;
10 tx=186:193;
11 ty=145:152;
12 ech = pastilles(ty, tx,:);
13 %figure, imshow(uint8(ech));
14 R=ech(:,:,1);
15 G=ech(:,:,2);
16 B=ech(:,:,3);
17
18 R=R(:)
19 G=G(:)
20 B=B(:)
21
22 r=mean(R,"all");
23 g=mean(G,"all");
24 b=mean(B,"all");
25 mu=[r,g,b]
26 S11 = sum((R-r).^2);
27 S12 = sum((R-r).*(G-g));
28 S13 = sum((R-r).*(B-b));
29 S21 = sum((G-g).*(R-r));
30 S22 = sum((G-g).*(G-g));
31 S23 = sum((G-g).*(B-b));
32 S31 = sum((B-b).*(R-r));
33 S32 = sum((B-b).*(G-g));
34 S33 = sum((B-b).*(B-b));
35
36 Sigma = [S11, S12, S13;
37           S21, S22, S23;
38           S31, S32, S33]
39 SigmaInv = inv(Sigma);
40 [lignes, cols, ~] = size(pastilles);
41 dist=zeros(lignes,cols);
42
43 for y = 1:cols
44     for x = 1:lignes
45         pixel = squeeze(pastilles(x, y, :));
46         diff = pixel - mu';
47         dist(x,y) = diff' * SigmaInv * diff;
48     end
49 end
50
51
52
53 M = dist >5 & dist < 61;
54 figure,imshow(uint8(M*255));
55 avant_plan=pastilles.*(1-M);
```

7.5 Changement de couleur des pastilles

```
1 %changement de couleur
2 avant_plan_hsv = rgb2hsv(avant_plan);
3 color = 0.2;
4 avant_plan_hsv(:, :, 1) = color;
5 avant_plan_hsv(:, :, 2) = 1;
6 avant_plan_rgb = hsv2rgb(avant_plan_hsv);
7 arriere_plan=pastilles.*M;
8 arriere_plan = double(arriere_plan)/255;
9 final = avant_plan_rgb + arriere_plan;
10 figure, imshow(final);
```